

UNIVERSIDAD POLITÉCNICA SALESIANA

ESCUELA DE INGENIERÍA

**DESARROLLO DE UN SISTEMA HMI / SCADA, DISEÑADO
CON LÓGICA DIFUSA E IMPLEMENTADO EN UN PROTOTIPO
DE PRUEBA.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
ESPECIALISTA EN SISTEMAS
MENCIÓN ROBÓTICA E INTELIGENCIA ARTIFICIAL**

CARRANCO QUIÑÓNEZ CARLOS ALBERTO

CELI SÁNCHEZ CARMEN JOHANNA

DIRECTOR: ING. CARLOS PILLAJO ANGOS

Quito, Enero 2010

DECLARACIÓN

Nosotros, Carranco Quiñónez Carlos Alberto y Celi Sánchez Carmen Johanna, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

CARRANCO QUIÑÓNEZ CARLOS ALBERTO.

CELI SÁNCHEZ CARMEN JOHANNA.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Carranco Quiñónez Carlos Alberto y Celi Sánchez Carmen Johanna, bajo mi supervisión.

Ing. Carlos Pillajo Angos.
DIRECTOR DE PROYECTO

DEDICATORIA

A mis padres Julia y Carlos por el esfuerzo incondicional que han realizado por sus hijos y hoy pueden verlo reflejado, porque hemos logrado alcanzar grandes éxitos, tanto personales como profesionales, nos dieron grandes valores, para seguir creciendo y siempre ser los mejores.

A mis hermanos Darío, Miguel, David por ser buenos amigos. Que la amistad que tenemos día a día siga creciendo.

Carlos Alberto Carranco Quiñónez

AGRADECIMIENTOS

Primeramente a Dios por brindarme la oportunidad de finalizar mis estudios y conocer las personas indicadas, que me han apoyado en el transcurso de mi vida, tanto personal como profesional.

A mi tutor de tesis Ing. Carlos Pillajo Angos, por confiar en nosotros y brindarnos su apoyo y la oportunidad de crecer profesionalmente.

A mí querida amiga Johanna Celi Sánchez, porque me brindaste tu amistad desde el inicio de nuestros estudios y nunca me has fallado como amiga, el haber formado un buen equipo de trabajo y poder lograr finalizar nuestra tesis con éxito. Te deseo lo mejor en tu vida.

A la familia Quiñónez Olives y familia Carranco Lozada, por el amor y cariño que tienen hacia mí, siempre han estado en los buenos y malos momentos apoyándome.

Carlos Alberto Carranco Quiñónez

DEDICATORIA

Quiero dedicar mi tesis de grado a mis progenitores, a mi padre Edwin Celi que no se encuentra entre nosotros, la voluntad de Dios ha sido llevarlo a su lado, pero estoy segura que si habitara aquí estaría muy orgulloso de mí. Padre siempre te recordaré.

A mi mamita querida Carmen Sánchez, que siempre me ha motivado a salir adelante y ha sido una persona excepcional, con su amor y perspicacia ha realizado un buen trabajo con sus hijos, es difícil poder expresar con palabras todo el inmenso amor que siento hacia mi señora linda. Madre muchas gracias por todo.

A mi esposo Henry Alvarez, que me ha apoyado en todos los aspectos de mi vida, con su ayuda, amor y comprensión puedo luchar día a día y vencer los obstáculos que se me presentan, ha sido mi inspiración para seguir adelante. Gracias mi amor por compartir tu vida conmigo.

A mis hermanas, Kelly y Gina Celi, que siempre están dispuestas y comprometidas con la familia, por el soporte emocional y económico siempre incondicional que me han brindado. Gracias hermanitas por su sagacidad e inteligencia porque ustedes han sido la clave de mi éxito.

A mi hermano Edwin Celi, aunque a la distancia, yo sé que me ha ayudado con todas las fuerzas de su corazón, su sustento afectivo y económico ha colaborado para que yo pueda obtener mi ingeniería.

A Henry Carrión, Ángel Jiménez, Jackeline Celi y a todos mis sobrinos, porque su rol dentro de la familia ha sido una influencia importante para yo sea una persona integra. Ustedes han sido un ejemplo de superación.

Carmen Johanna Celi Sánchez.

AGRADECIMIENTOS

Primero quiero agradecer a mi Dios por permitirme cumplir con mis objetivos, me ha guiado en todo momento brindándome estabilidad emocional. Ha sido esencial en mi vida, me ha dado toda la sabiduría y protección que he necesitado.

Deseo expresar mis sinceras gratitudes, a la Universidad Politécnica Salesiana, porque con su diaria labor ayuda a que los estudiantes podamos alcanzar el sueño de ser profesionales.

Agradezco al Ing. Carlos Pillajo, mi director de tesis, por guiarme con sus consejos profesionales para la realización de este trabajo de investigación y principalmente por compartir desinteresadamente sus amplios conocimientos y experiencias.

También quiero agradecer a mi amigo Carlitos Carranco, mi compañero de tesis, ya que sin él no hubiese sido posible culminar el proyecto, gracias por brindarme tu sincera amistad y espero seguir siendo buenos amigos en la nueva vida que emprendemos como profesionales.

Debería escribir una lista inmensa de los excelentes profesores que me han instruido en mi vida estudiantil, pero me faltarían hojas para describir lo agradecida que me siento con ellos. Gracias a todos por enseñarme a ser una persona constituida moral y profesionalmente.

Carmen Johanna Celi Sánchez.

CONTENIDO

RESÚMEN.....	1
PRESENTACIÓN.....	4
CAPÍTULO 1. LÓGICA DIFUSA.....	7
1.1. FUSIFICACIÓN.....	8
1.1.1. CONJUNTOS DIFUSOS.....	8
1.1.1.1. Función de Membresía.....	9
1.1.1.2. Tipos de Conjuntos Difusos.....	9
1.1.1.3. Operaciones con Conjuntos Fuzzy.....	10
1.1.2. VARIABLE LINGÜÍSTICA.....	15
1.2. INFERENCIA.....	16
1.2.1. REGLAS LÓGICAS.....	17
1.2.2. MEMORIA ASOCIATIVA FUZZY.....	18
1.2.3. RELACIONES.....	19
1.2.4. REGLAS DE INFERENCIA.....	20
1.3. DESFUSIFICACIÓN.....	27
1.3.1. PUNTO MÁXIMO.....	27
1.3.2. CENTRO DE ÁREA.....	30
1.3.3. CENTRO DE GRAVEDAD.....	32
1.4. MODELOS DIFUSOS.....	33
1.4.1. DEFINICIÓN DE CARACTERÍSTICAS.....	34
1.4.2. DEFINICIÓN DE CONJUNTOS DIFUSOS.....	36
1.4.3. DEFINICIÓN DE REGLAS.....	41
1.4.3.1. Diseño de una base de reglas.....	42
1.4.4. ELECCIÓN DEL MÉTODO DE DESFUSIFICACIÓN.....	44
1.4.5. SIMULACIÓN Y AJUSTE DEL SISTEMA.....	46
1.5. APLICACIONES DE LÓGICA DIFUSA.....	53
CAPÍTULO 2. SISTEMAS HMI / SCADA.....	57
2.1. PROGRESO DE LOS SISTEMAS HMI/SCADA.....	60
2.2. COMPONENTES DE UN HMI / SCADA.....	62
2.2.1. UNIDAD TERMINAL REMOTA.....	62
2.2.2. ESTACIÓN MAESTRA.....	63
2.2.3. INFRAESTRUCTURA DE COMUNICACIÓN.....	64

2.3.	PRESTACIONES DE UN HMI / SCADA.	65
2.4.	REQUISITOS DE UN HMI / SCADA.	66
2.5.	MÓDULOS DE UN HMI / SCADA.	66
2.6.	CONCEPTOS ASOCIADOS A UN HMI / SCADA.	67
2.6.1.	TIEMPO REAL.	67
2.6.2.	HARDWARE EN SISTEMAS DE SUPERVISIÓN.	67
2.6.3.	SOFTWARE DE LOS SISTEMAS HMI / SCADA.	68
2.6.4.	TARJETAS DE EXPANSIÓN.	69
2.6.5.	ESTRUCTURA ABIERTA.	70
2.6.6.	COMPUTADORES INDUSTRIALES.	70
2.7.	AUTÓMATA LÓGICO PROGRAMABLE.	72
2.7.1.	FUNCIONAMIENTO.	73
2.7.1.1.	Proceso Inicial.	74
2.7.1.2.	Ciclo de operación.	75
2.7.2.	FUNCIONES DEL PLC.	76
2.7.3.	UNIDADES FUNCIONALES.	76
2.7.3.1.	Unidad de Entradas.	77
2.7.3.2.	Unidad de Salidas.	77
2.7.3.3.	Unidad Lógica.	77
2.7.3.4.	Unidad de memoria.	77
2.7.4.	MÓDULOS DE UN PLC.	78
2.7.4.1.	Módulo de comunicaciones.	78
2.7.4.2.	Módulo de control de redundancia.	79
2.7.4.3.	Módulo para conexión de rack remoto.	79
2.7.4.4.	Módulo de interfaz hombre-máquina.	79
2.7.4.5.	Módulos de Entrada y Salida.	80
2.7.5.	PROGRAMADOR / MONITOR (PM).	81
2.7.6.	VENTAJAS Y DESVENTAJAS DEL PLC.	81
2.7.7.	APLICACIONES DE LOS PLC.	83
2.7.8.	REDES INDUSTRIALES.	84
2.7.8.1.	Protocolos de Comunicación.	85
2.7.8.2.	Comunicación.	89
2.8.	INSTRUMENTACIÓN.	93

2.8.1.	DEFINICIONES BÁSICAS.....	93
2.8.2.	APLICACIÓN EN LA INDUSTRIA.	94
2.8.3.	SÍMBOLOS DE INSTRUMENTACIÓN.....	95
2.8.3.1.	Símbolos y Números de Instrumentación.	95
2.8.4.	IDENTIFICACIÓN FUNCIONAL.....	99
2.8.5.	IDENTIFICACIÓN Y DIAGRAMA DEL LAZO.....	101
2.8.5.1.	Set Point y Rango de Operación.	101
2.8.6.	LAZOS ELECTRÓNICOS.	102
CAPÍTULO 3. HARDWARE		105
3.1.	SENSORES.....	105
3.1.1.	ÁREAS DE APLICACIÓN DE LOS SENSORES.....	106
3.1.2.	CARACTERÍSTICAS DE UN SENSOR.	106
3.1.3.	RESOLUCIÓN Y PRECISIÓN.	107
3.1.4.	TIPOS DE SENSORES.....	107
3.1.5.	SENSOR DE TEMPERATURA.....	109
3.1.6.	SENSOR DE LUZ.....	112
3.2.	ACTUADORES.....	114
3.2.1.	VENTILADORES.....	114
3.2.2.	CALEFACTOR.....	115
3.2.3.	LÁMPARA.....	116
3.3.	DIAGRAMAS.	117
3.3.1.	DIAGRAMA DE COMPONENTES Y DISTRIBUCIÓN.....	117
3.3.2.	DIAGRAMA DE CASO DE USOS.....	120
3.3.3.	DIAGRAMA DE ESTADOS.....	121
3.3.4.	DIAGRAMA DE SECUENCIA.....	124
3.3.5.	DIAGRAMA DE BLOQUES.....	126
3.3.6.	DIAGRAMA DE CONECTIVIDAD.....	126
3.4.	PLC SIEMENS CPU 224.	129
3.4.1.	CARACTERÍSTICAS.....	131
3.4.2.	COMPONENTES.....	132
3.4.3.	CABLE PC/PPI.....	133
3.4.4.	CONCEPTOS BÁSICOS.....	134
3.4.5.	CONEXIÓN DE ENTRADAS Y SALIDAS.....	136

3.4.6.	DATOS TÉCNICOS DE LA CPU 224.....	138
3.5.	MÓDULO ANALÓGICO EM-235.....	140
3.5.1.	CARACTERÍSTICAS DEL MÓDULO ANALÓGICO EM-235.....	142
3.6.	PROTOTIPO DE PRUEBA.	146
CAPÍTULO 4. SOFTWARE		149
4.1	DIAGRAMAS.	150
4.1.1	DIAGRAMA DE COMPONENTES Y DISTRIBUCIÓN.....	151
4.1.2	DIAGRAMA DE ACTIVIDADES.....	154
4.1.3	DIAGRAMA DE ESTADO.....	155
4.1.4	DIAGRAMA DE SECUENCIA.....	157
4.2	LOOKOUT.....	159
4.2.1	INTERFAZ DE USUARIO.....	165
4.3	LABVIEW.....	166
4.3.1	VENTAJAS DE LABVIEW.....	170
4.3.2	INTERFAZ DE USUARIO.....	171
4.3.3	DIAGRAMA DE BLOQUES DE LABVIEW.....	174
4.4	STEP 7	178
4.4.1	CONCEPTOS BÁSICOS KOP.	182
4.4.2	PROGRAMA KOP.....	186
CAPÍTULO 5. ANÁLISIS DEL SISTEMA.....		202
5.1.	ANÁLISIS TÉCNICO ECONÓMICO.....	202
5.2.	MANUAL DE USUARIO.....	203
5.3.	PRUEBAS.....	223
5.4.	CONCLUSIONES Y RECOMENDACIONES.	229
5.4.1.	CONCLUSIONES.	229
5.4.2.	RECOMENDACIONES.....	231
REFERENCIAS BIBLIOGRÁFICAS		232
ANEXOS.....		238
ANEXO 1. VISUALIZADOR DE DATOS.....		238
ANEXO 2. FOTOGRAFÍAS DEL SISTEMA		246

RESÚMEN

Se desarrolla un sistema HMI/Scada, diseñado con Lógica Difusa e implementado en un prototipo de prueba.

El HMI/Scada con respecto al hardware, consta de un PLC, un módulo de extensión, una computadora y un prototipo de prueba.

El PLC es un Simatic S7-200 Siemens de CPU 224, debido a que el autómatas recibe entradas digitales y transmite salidas también digitales tipo relé, se realiza el acoplamiento con un módulo de extensión analógico EM235 que posee 4 entradas y 1 salida, para lograr el procesamiento de los datos en forma analógica. La computadora se utiliza para efectuar la supervisión y el control de los procesos. El prototipo de prueba se compone de una casa tipo invernadero, en donde están montados los sensores y actuadores, con respecto a los sensores tenemos: sensor de temperatura Termistor NTC y sensor de luz Fotorresistencia; por parte de los actuadores: Ventiladores para extraer el aire caliente de adentro hacia afuera, Calefactor para proporcionar calor, Ventilador de calefactor para empujar el aire al interior, y lámpara halógena. El prototipo nos permite realizar las pruebas respectivas para demostrar que la Lógica Difusa es aplicable en los sistemas HMI/Scada.

Se establecen las variables controladas: Temperatura con su Set Point a 24°C y Luminosidad con su Set Point al nivel 5.

El HMI/Scada con relación al software, se constituye de dos partes, la primera cuando el proceso de Lógica Difusa se realiza en la computadora y la segunda cuando se efectúa en el PLC. Se organiza de esta forma debido a que, cuando el autómatas está en laboratorio únicamente transmite y recibe datos porque la computadora controla y supervisa el sistema, y cuando el PLC está en campo debe ejecutar el proceso difuso para trabajar sin supervisión y sin el control de la PC.

NO se debe realizar Lógica Difusa en ambas partes (PC y PLC), esto origina que las variables se procesen dos veces y se obtengan datos erróneos o falsos.

Para el primer caso tenemos:

- LOOKOUT, sólo realiza supervisión con animaciones, se encuentra guardado en la PC. "Lectura de datos".
- LABVIEW, desarrollo del proceso de Lógica Difusa, en tres ventanas (Entradas, Reglas y Salidas), conexión con Lookout y con el PLC, se encuentra almacenado en la PC. "Lectura de datos de sensores y escritura de datos de actuadores".
- STEP7, configura el protocolo Freeport, se comunica con la computadora por el puerto serial, se encuentra almacenado en la memoria del PLC. "Envío de datos de sensores y recepción de datos de actuadores".

Para el segundo caso tenemos:

- LOOKOUT, la misma interfase anteriormente mencionada.
- LABVIEW, interfaz de comunicación entre el software Lookout y el autómata, almacenado en la computadora. "Lectura de datos".
- STEP7, configura el protocolo Freeport para comunicarse con la computadora, desarrolla el proceso de Lógica Difusa, se encuentra cargado en el PLC. "Transmisión de datos".

El sistema está basado en lógica difusa con el fin de obtener respuestas precisas y más acordes con la realidad. Los sistemas HMI/Scada se construyen con orígenes en la lógica clásica, la cual no brinda resultados completos, debido a que trabaja sólo con 2 estados por ejemplo:

- Estado alto y estado bajo (High y Low),
- 0 lógico y 1 lógico,
- Encendido y apagado,
- 0 voltios y 5 voltios,
- 0 voltios y 12 voltios,
- 0 voltios y 24 voltios, etc.

El presente proyecto de tesis se crea a partir de una Interfaz Hombre Máquina, ya que un Scada se trata de una red de autómatas, y un HMI es una parte del Scada estudiando esencialmente la supervisión a través de un software en una computadora con la conexión a un PLC. Entonces podemos decir que el conjunto de HMI's forman un sistema Scada completo.

PRESENTACIÓN

PLANTEAMIENTO DEL PROBLEMA:

La propuesta es elaborar un sistema basado en lógica difusa que responda de una manera más precisa, real y con muchos valores aceptables. Los sistemas Scada se construyen con orígenes en la lógica clásica la cual no brinda resultados completos, debido a que trabaja sólo con dos estados por ejemplo cero y uno, encendido y apagado, 0 voltios y 5 voltios, 0 voltios y 12 voltios, etc.

La respuesta de lógica difusa debe ser interpretada por un autómata programable que opere con señales analógicas, debido a que los PLC's poseen entradas digitales y salidas tipo relé, se debe incluir un módulo analógico al autómata.

Además se necesita un software orientado al operador o usuario, para que realice el control del sistema, el programa debe trabajar en tiempo real, y permitir la supervisión de señales. También se debe crear un prototipo para efectuar las pruebas, que permitan demostrar la efectividad de la lógica difusa en los sistemas HMI/Scada.

OBJETIVOS.

OBJETIVO GENERAL:

- Desarrollar un sistema HMI / SCADA, diseñado con Lógica Difusa e implementado en un prototipo de prueba, el cual permitirá al usuario el control y la supervisión de todo el proceso.

OBJETIVOS ESPECÍFICOS:

- Encontrar información provechosa y relevante a Lógica Difusa, sistemas HMI / Scada, autómatas programables y módulo de extensión analógico.
- Diseñar de una forma clara y precisa las diferentes entradas y salidas difusas que se van a utilizar en el proceso de fusificación, para alcanzar la respuesta que mejor se adapte a la realidad y elegir el mejor método de defusificación.
- Definir los sensores y actuadores más convenientes, para su aplicación en el presente sistema.

- Crear el hardware necesario (prototipo) para realizar las pruebas de adquisición de señales y mostrar al usuario el proceso del sistema.
- Programar el proceso de lógica difusa en el lenguaje compatible con el autómata que se va a adquirir dependiendo de la marca.
- Lograr una correcta comunicación entre el software y el hardware especialmente con el módulo de extensión análogo y con los sensores y actuadores del prototipo de prueba.
- Supervisar el comportamiento del hardware a través del software para el HMI / Scada en tiempo real.

HIPÓTESIS:

Mediante el sistema a implementarse se puede demostrar que la Lógica Difusa es efectiva para los sistemas HMI / Scada.

JUSTIFICACIÓN:

- a) *Justificación Teórica:* La gran ventaja de utilizar el proceso teórico de Lógica Difusa, es que se puede dotar al sistema de la capacidad en su respuesta no ambigua entre 2 estados, que son limitantes para la acción de los sensores y actuadores.
- b) *Justificación Metodológica:* El manejo de un autómata programable beneficia algunos aspectos como: la modernidad en los sistemas de control, la ejecución remota de comandos, la adquisición de datos en tiempo real, trabaja en forma rápida y segura, almacenamiento de datos, ocupan menos espacio físico, etc. A esto le incluimos la capacidad del módulo de extensión, el cual permite manejar señales analógicas.
- c) *Justificación Práctica:* El ser humano necesita de sistemas que respondan con mayores alternativas, a diferencia de los dispositivos comunes que en su mayoría no las brindan, por ejemplo aparatos encendidos o apagados. Las personas podemos percibir la diferencia entre varias respuestas así el nivel de temperatura de un calefactor podría ser: extra alto, alto, medio, bajo, extra bajo etc.

ALCANCE:

Se inicia con el diseño del proceso de Lógica Difusa, posteriormente se implementa con el hardware necesario, a continuación se programa el proceso difuso en el software conveniente, luego se lleva a cabo la adquisición de los datos, consecutivamente se realiza el control y la supervisión el sistema y por último se ejecuta las pruebas.

Al realizar las pruebas con la ayuda del prototipo, se analiza el comportamiento de los sensores y actuadores analógicos, con las respectivas perturbaciones creadas artificialmente por nosotros.

METODOLOGÍA:

Partiendo del estudio descriptivo se puede desarrollar el software de Lógica Difusa, identificando características de los conjuntos universo difuso de cada entrada y de cada salida, señalando formas de conducta en cada uno de los subconjuntos.

Con este estudio se consigue diseñar las entradas, salidas y variables lingüísticas de una forma concreta y precisa representando de la mejor manera al proceso real. Asimismo permite descubrir el mejor método de defusificación. Se realiza un programa de supervisión pensado en el usuario final, con una interfaz amigable.

Por medio del estudio exploratorio se consigue información sobre los autómatas, su configuración, el software que requieren para su programación, el manejo de los módulos externos analógicos, etc. Además información necesaria para el desarrollo de Lógica Difusa y de todos los lenguajes a utilizar en el sistema.

El hardware debe complementarse con los sensores, actuadores y el prototipo de prueba, además con perturbaciones artificiales para evidenciar los resultados esperados. Esto se puede conseguir con el estudio explicativo.

CAPÍTULO 1. LÓGICA DIFUSA

Lógica es un término que se deriva del griego “logos” que significa “razón”¹. Es el estudio científico del proceso de razonamiento, es la ciencia que estudia las condiciones formales de validez de una inferencia y en general de cualquier proposición.

El término “fuzzy” fue creado por el Profesor Emeritus de Columbia University: Ph.D Lotfi Zadeh (1921). *“Yo estaba consciente del hecho de que podría ser controversial, pero no podía pensar en otro término más respetable para describir lo que tenía en mente, que eran conjuntos que no tenían límites exactos, como nubes.”*². Lotfi intentó demostrar la inteligencia de las máquinas para interpretar valores lógicos ambiguos o sentencias imprecisas que pueden pertenecer a varios contextos.

La lógica difusa es una rama de la Inteligencia Artificial que surge como un intento de formalizar el conocimiento que presenta incertidumbre, indecisión, ambigüedad o vaguedad. Aborda problemas definidos en términos lingüísticos y por tanto imprecisos, donde la información esta expresada en términos cualitativos.

Lógica difusa es un tipo de lógica que reconoce más que simples valores verdaderos y falsos; con ella las proposiciones pueden ser representadas con grados de veracidad o falsedad. Por ejemplo: la sentencia "hoy es un día soleado", puede ser 100% verdad si no hay nubes, 80% verdad si hay pocas nubes, 50% verdad si hay muchas nubes, 20% verdad si existe neblina y 0% si llueve todo el día³.

El término fuzzy, se encuentra en nuestras decisiones, en nuestro pensamiento, en la forma cómo procesamos la información, pero especialmente en nuestro lenguaje, ya que en muchas ocasiones no expresamos con claridad lo que queremos comunicar. Frases como: "Nos vemos luego", "un poco más", "no me siento muy bien", son expresiones difusas. En ocasiones, enunciados difusos indican unidades relativas y subunidades más no unidades absolutas.

Un sistema Fuzzy consta de los siguientes bloques:

¹ <http://pochicasta.files.wordpress.com/2008/01/definicion-de-la-logica.pdf> Origen griego de lógica creado por Aristóteles.

² <http://kim.ece.buap.mx/posgrado/Fuzzytutorial/zadeh.htm> Biografía de Lotfi Zadeh (Iraní 1921).

³ <http://personales.ya.com/casanchi/mat/difusa01.htm> ¿Qué es lógica difusa?

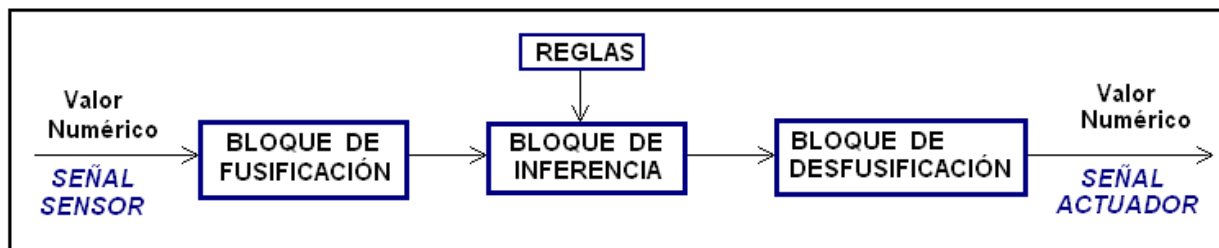


Figura 1.1 Proceso Difuso.

1.1. FUSIFICACIÓN.

Fusificar es tomar una señal continua en la entrada (un valor numérico o señal de un sensor) y transformarla en un conjunto de valores entre 0 y 1 que indiquen el grado de pertenencia (la función de membresía) en cada conjunto fuzzy establecido.

En este bloque primero se debe estudiar la teoría de conjuntos difusos, para poder establecer el diseño y la forma que van a tomar las variables de entrada y de salida del proceso.

1.1.1. CONJUNTOS DIFUSOS.

Un conjunto difuso es una forma de representar los fenómenos de la realidad, los cuáles no sólo tienen dos valores como Verdadero o Falso, sino que tienen un infinito número de estados comprendidos entre 0 y 1, es por tal motivo que en un conjunto difuso cada uno de sus elementos toma un grado de pertenencia en dicho conjunto, esta función es la que se encarga de ligar los elementos del conjunto U universo de discurso con los elementos del intervalo entre 0 y 1.

Cada uno de estos elementos puede ser miembro de varios conjuntos difusos pero cada uno con un grado de pertenencia.

Un conjunto es la agrupación de elementos con una característica común.

La simbología de conjuntos que se utilizará es:

Universo de discurso	→	U	Contiene todos los elementos.
Vacío	→	\emptyset	No contiene ningún elemento.
Unitario	→	\hat{o}	Sólo contiene 1 elemento.

Los conjuntos se pueden representar de varias formas como por ejemplo por Extensión y Comprensión, en lógica difusa por lo general se representan por medio de la función de membresía.

Ejemplos:

$$A = \{3, 4, 5, 6, 7, 8, 9\}$$

Conjunto A representado por Extensión.

$$M = \{x \in \mathbb{N}^+ / x \geq 3 \wedge x \leq 9\}$$

Conjunto M representado por Comprensión.

1.1.1.1. Función de Membresía.

La función de membresía es el grado de pertenencia que tienen los elementos del conjunto U de discurso, son valores entre 0 y 1.

La simbología de la función de membresía es la siguiente: $\mu_A(x)$.

En el siguiente ejemplo básico, la función de membresía se conforma por 2 tipos de valores 0 y 1 pero en lógica difusa comprende una infinidad de valores:

$$\mu_G(x) = \begin{cases} 1 & 2 \leq x \leq 9 \\ 0 & \text{en otro caso} \end{cases}$$

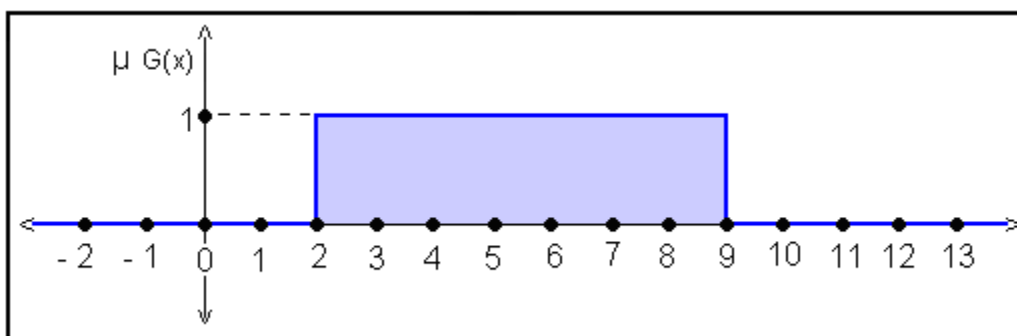


Figura 1.2 Gráfico del ejemplo de función de membresía.

1.1.1.2. Tipos de Conjuntos Difusos.

Los conjuntos difusos pueden ser de varios tipos, pero siempre deben ser simétricos. Entre algunos ejemplos tenemos: Triangular, Trapezoidal, Campana de Gauss, Sigmoidal, Unitario, etc. Siendo los más usados el triangular y trapezoidal. El valor del máximo invariablemente es 1.

En la siguiente figura se detalla el gráfico de cada conjunto:

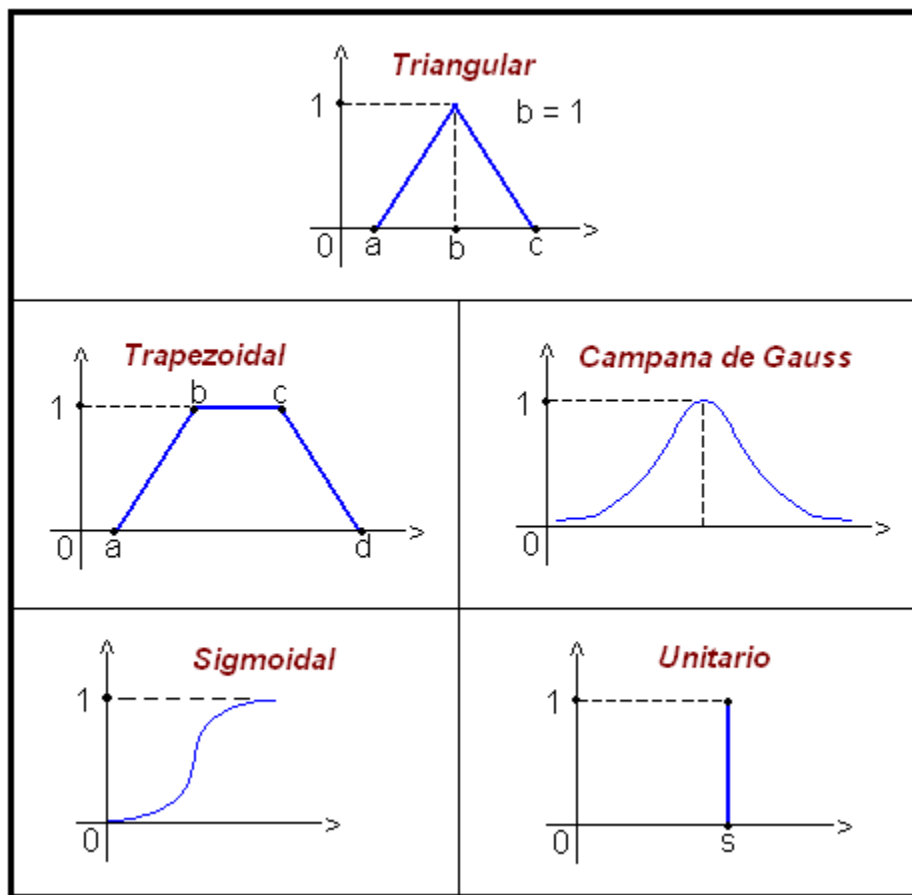


Figura 1.3 Tipos de conjuntos difusos.

1.1.1.3. Operaciones con Conjuntos Fuzzy.

Entre las operaciones más importantes están: Intersección, Unión y Complemento.⁴

Intersección:

La Intersección se simboliza: $\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x)$.

Tipos de Intersección:

a) Producto Algebraico: $\mu_A * \mu_B = \mu_A \cdot \mu_B$

b) Mínimo: $\mu_A * \mu_B = \min(\mu_A, \mu_B)$

c) Producto Drástico:
$$\mu_A * \mu_B = \begin{cases} \mu_A & \text{si } \mu_B = 1 \\ \mu_B & \text{si } \mu_A = 1 \\ 0 & \text{en otro caso} \end{cases}$$

⁴ Lotfi A. Zadeh desarrolló una algebra completa sobre conjuntos difusos en su publicación: **Fuzzy Sets** "Information and Control", Vol. 8, 1965, pp. 338-353.

d) Producto Einstein:
$$\mu_A * \mu_B = \frac{\mu_A \cdot \mu_B}{2 - (\mu_A + \mu_B - \mu_A \cdot \mu_B)}$$

Unión:

La Unión se simboliza: $\mu_A + \mu_B$

Tipos de Unión:

a) Suma Algebraica:
$$\mu_A \oplus \mu_B = \mu_A + \mu_B - \mu_A \cdot \mu_B$$

b) Máximo:
$$\mu_A + \mu_B = \max(\mu_A, \mu_B)$$

c) Suma Drástica:
$$\mu_A \oplus \mu_B = \begin{cases} \mu_A & \text{si } \mu_B = 0 \\ \mu_B & \text{si } \mu_A = 0 \\ 1 & \text{en otro caso} \end{cases}$$

d) Suma Einstein:
$$\mu_A \oplus \mu_B = \frac{\mu_A + \mu_B}{1 + \mu_A + \mu_B}$$

e) Bounded Sum (Suma Limitada):
$$\mu_A \oplus \mu_B = \min(1, \mu_A + \mu_B)$$

Complemento:

El Complemento se simboliza y es: $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Para un mejor entendimiento se realiza un ejemplo con la intersección, unión y complemento del siguiente ejercicio:

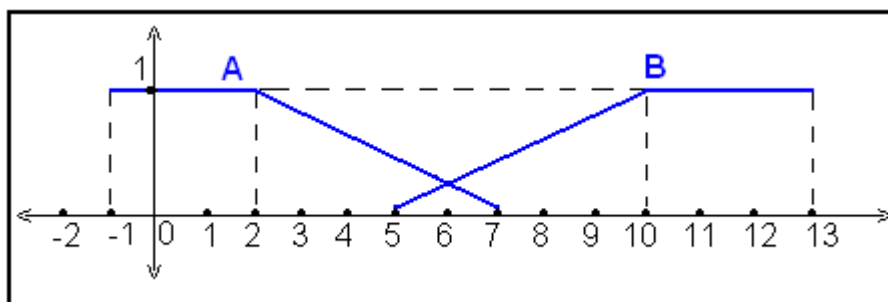


Figura 1.4 Gráfico base para las operaciones.

La Intersección se determina con el mínimo entre los valores, y la unión con el máximo y el complemento es 1 menos la función de membresía. El conjunto U de discurso es desde: -1 hasta 13.

Intersección:

$$\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \}$$

$$\mu_{A \cap B}(-1) = \min \{1, 0\} = 0$$

$$\mu_{A \cap B}(0) = \min \{1, 0\} = 0$$

$$\mu_{A \cap B}(2) = \min \{1, 0\} = 0$$

$$\mu_{A \cap B}(3) = \min \{0.8, 0\} = 0$$

$$\mu_{A \cap B}(4) = \min \{0.6, 0\} = 0$$

$$\mu_{A \cap B}(5) = \min \{0.4, 0\} = 0$$

$$\mu_{A \cap B}(6) = \min \{0.2, 0.2\} = 0.2$$

$$\mu_{A \cap B}(7) = \min \{0, 0.4\} = 0$$

$$\mu_{A \cap B}(8) = \min \{0, 0.6\} = 0$$

$$\mu_{A \cap B}(9) = \min \{0, 0.8\} = 0$$

$$\mu_{A \cap B}(10) = \min \{0, 1\} = 0$$

$$\mu_{A \cap B}(11) = \min \{0, 1\} = 0$$

$$\mu_{A \cap B}(12) = \min \{0, 1\} = 0$$

$$\mu_{A \cap B}(13) = \min \{0, 1\} = 0$$

Unión:

$$\mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \}$$

$$\mu_{A \cup B}(-1) = \max \{1, 0\} = 1$$

$$\mu_{A \cup B}(0) = \max \{1, 0\} = 1$$

$$\mu_{A \cup B}(1) = \max \{1, 0\} = 1$$

$$\mu_{A \cup B}(2) = \max \{1, 0\} = 1$$

$$\mu_{A \cup B}(3) = \max \{0.8, 0\} = 0.8$$

$$\mu_{A \cup B}(4) = \max \{0.6, 0\} = 0.6$$

$$\mu_{A \cup B}(5) = \max \{0.4, 0\} = 0.4$$

$$\mu_{A \cup B}(6) = \max \{0.2, 0.2\} = 0.2$$

$$\mu_{A \cup B}(7) = \max \{0, 0.4\} = 0.4$$

$$\mu_{A \cup B}(8) = \max \{0, 0.6\} = 0.6$$

$$\mu_{A \cup B}(9) = \max \{0, 0.8\} = 0.8$$

$$\mu_{A \cup B}(10) = \max \{0, 1\} = 1$$

$$\mu_{A \cup B}(11) = \max \{0, 1\} = 1$$

$$\mu_{A \cup B}(12) = \max \{0, 1\} = 1$$

$$\mu_{A \cup B}(13) = \max \{0, 1\} = 1$$

Complemento de A:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

$$\mu_{\bar{A}}(-1) = 1 - 1 = 0$$

$$\mu_{\bar{A}}(0) = 1 - 1 = 0$$

$$\mu_{\bar{A}}(1) = 1 - 1 = 0$$

$$\mu_{\bar{A}}(2) = 1 - 1 = 0$$

$$\mu_{\bar{A}}(3) = 1 - 0.8 = 0.2$$

$$\mu_{\bar{A}}(4) = 1 - 0.6 = 0.4$$

$$\mu_{\bar{A}}(5) = 1 - 0.4 = 0.6$$

$$\mu_{\bar{A}}(6) = 1 - 0.2 = 0.8$$

$$\mu_{\bar{A}}(7) = 1 - 0 = 1$$

Complemento de B:

$$\mu_{\bar{B}}(x) = 1 - \mu_B(x).$$

$$\mu_{\bar{B}}(5) = 1 - 0 = 1$$

$$\mu_{\bar{B}}(6) = 1 - 0.2 = 0.8$$

$$\mu_{\bar{B}}(7) = 1 - 0.4 = 0.6$$

$$\mu_{\bar{B}}(8) = 1 - 0.6 = 0.4$$

$$\mu_{\bar{B}}(9) = 1 - 0.8 = 0.2$$

$$\mu_{\bar{B}}(10) = 1 - 1 = 0$$

$$\mu_{\bar{B}}(11) = 1 - 1 = 0$$

$$\mu_{\bar{B}}(12) = 1 - 1 = 0$$

$$\mu_{\bar{B}}(13) = 1 - 1 = 0$$

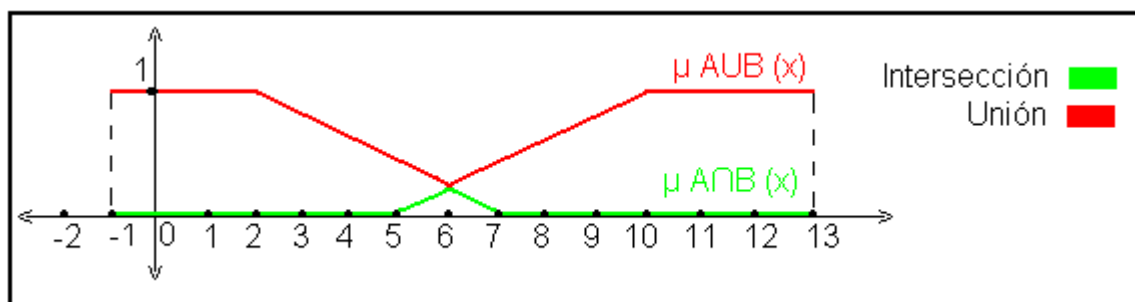


Figura 1.5 Respuesta gráfica de intersección y unión.

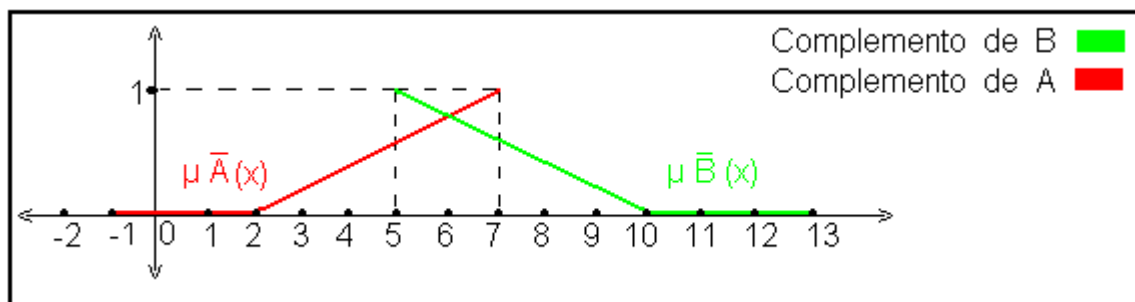


Figura 1.6 Respuesta gráfica de complementos.

Para poder calcular el valor de la función de membresía, utilizamos la fórmula de la pendiente:

$$y = \frac{(y_2 - y_1)}{(x_2 - x_1)} * (x - x_1) + y_1$$

1.1.2. VARIABLE LINGÜÍSTICA.

La variable Lingüística permite etiquetar a cada uno de los conjuntos expresados y define al conjunto universo. Refleja el funcionamiento del sistema representando el lenguaje natural. Sus características⁵ son:

N = nombre de la variable. Ejemplo: caluroso, frío, alto, bajo, moderado, etc.

V = conjunto de valores que la variable puede tomar. Ejemplo: 5 – 33.

U = universo de discurso en el que se evalúa la variable. Ejemplo: 0 – 100.

S = regla semántica asignada a un conjunto fuzzy en el universo para cada valor lingüístico en V .

Un ejemplo completo a continuación:

N : edades.

V : niño, adolescente, adulto, anciano.

U : 0 – 90 años.

S : μ niño (x), μ adolescente (x), μ adulto (x), μ anciano (x).

Existe un estándar de conjuntos con sus variables lingüísticas, pero eso no significa que el diseñador del sistema fuzzy debe regirse obligatoriamente a él, este estándar es una forma optimizada de diseño pero no siempre es la apropiada:

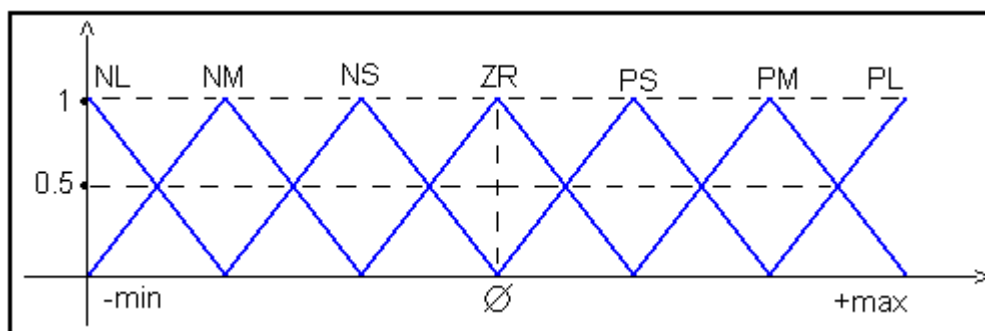


Figura 1.7 Estándar de Conjuntos.

NL = Negative Large. Valor más alto en el rango negativo.

NM = Negative Medium. Valor negativo medio.

NS = Negative Small. Valor negativo pequeño.

⁵ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/2var_ling.html
lingüísticas.

ZR = Zero. Valor Neutral.

PS = Positive Small. Valor positivo pequeño.

PM = Positive Médium. Valor positivo medio.

PL = Positive Large. Valor más alto en el rango positivo.

1.2. INFERENCIA.

Inferir es la capacidad de construir conclusiones apropiadas, frente a una evidencia dada, es decir se puede ver a la inferencia como un proceso que va desde lo que conocemos (una premisa) hasta lo que no conocemos (conclusión).

La Inferencia es una forma elegante de referirse a la búsqueda, también se la puede llamar máquina o motor de búsqueda.

Por tanto la Inferencia es un motor que tiene la capacidad de decisión.

La Inferencia se consigue a través del Conocimiento. Una Base de Conocimientos consiste en reglas fuzzy de tipo If, Then y funciones de membresía caracterizadas por los conjunto difusos.

Para el diseño de la base de conocimientos donde se encuentran las reglas de control, hay que tomar en cuenta: primero, que el conjunto de variables lingüísticas seleccionadas debe escogerse de forma tal que describan muy bien a los parámetros de control del proceso.

Tanto los parámetros de entrada como los de salida deben definirse en esta etapa empleando terminología apropiada.

La selección del rango de valores para cada término de las variables de entrada y salida es muy importante para lograr la "*suavidad*" del control.

En segundo lugar la base de conocimientos se elabora tomando en cuenta la descripción lingüística⁶ de los parámetros, para ello se sugieren cuatro métodos:

- a) Experiencia y conocimiento de un operador humano.
- b) Modelado de las acciones de control del operador.
- c) Modelado del proceso.
- d) Auto-sintonización.

⁶ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/3basede_conoc.html Parámetros que se deben tomar en cuenta para la elaboración de la base de conocimientos.

De los métodos anteriores, el primero es el más utilizado. La efectividad del método se basa en que el operador puede expresar los conocimientos que utiliza en el control del procesos en términos de reglas lógicas.

El segundo método modela directamente la acción de control del operador.

El tercer método trata con el modelo difuso del proceso, aquí se configura el modelo de una planta como una aproximación utilizando implicaciones que describen los posibles estados del sistema. En este método se desarrolla un modelo y se construye un controlador difuso para controlar el modelo difuso, lo descrito anteriormente es similar a lo que se hace en teoría de control clásica. Aquí se necesita identificar la estructura y los parámetros del proceso.

La idea principal del cuarto método es el desarrollo de reglas que pueden ajustarse aún y cuando el controlador este en uso para lograr mejorar su desempeño.

De todos los métodos que se pueden utilizar para construir la base de conocimientos, en el presente trabajo recurrimos al primero que se basa en reglas lógicas.

1.2.1. REGLAS LÓGICAS.

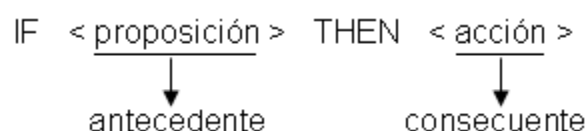
Una regla es un criterio de condición que representa el conocimiento. Cuando modelamos el conocimiento del operador las reglas de control toman la forma:

SI [estatura > 1.80 m] ENTONCES [persona es alta].

Esto en español, pero los sistemas informáticos soportan inglés:

IF [estatura < 1.60 m] THEN [persona es pequeña].

En general podemos decir:



Las reglas también se forman con operadores lógicos como son: AND, OR, NOT.

Los valores de verdad de los operadores son:

A	B	$A \wedge B$	C	D	$C \vee D$	E	$\sim E$
V	V	V	V	V	V	V	F
V	F	F	V	F	V	F	V
F	V	F	F	V	V		
F	F	F	F	F	F		
AND			OR			NOT	

A	B	$\min(A, B)$	C	D	$\max(C, D)$	E	$1 - E$
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		
Intersección $A \cap B$			Unión $C \cup D$			Complemento \bar{E}	

Figura 1. 8 Operadores lógicos y operaciones con conjuntos.

Unos ejemplos de reglas con operadores lógicos son:

IF estatura = alta AND #canastas < 30 THEN calidad _ jugador = media.

IF cantidad _ ropa = abundante OR nivel _ suciedad = alto THEN
cantidad _ detergente = moderado.

IF cielo = despejado THEN NOT lluvia.

1.2.2. MEMORIA ASOCIATIVA FUZZY.

La FAM⁷ constituye una matriz que permite asociar las reglas con las salidas.

Con la FAM conocemos las reglas que se van a activar, como también las relaciones conocidas entre los antecedentes y consecuentes.

La limitación de esta matriz es que sólo permite representar la consecuencia de cada regla definida por la combinación de dos entradas.

Podemos decir que la FAM es una forma gráfica de representar las relaciones entre los conjuntos fuzzy.

Un ejemplo es:

⁷ FAM “Fuzzy Associative Memory” en español: Memoria Asociativa Difusa.

		Velocidad		
Distancia		LENTA	MEDIA	RÁPIDA
	GRANDE	0%	0%	50%
	NORMAL	0%	50%	50%
	PEQUEÑA	50%	100%	100%

Salida = Freno %.

Figura 1. 9 Ejemplo FAM.

1.2.3. RELACIONES.

Una relación difusa representa el grado de presencia o ausencia de una asociación, interacción o interconexión entre elementos de dos o más conjuntos difusos, la relación se establece a partir del producto cartesiano entre los subconjuntos fuzzy.

Las relaciones nos permiten representar un subconjunto similar al subconjunto del producto cartesiano que está formado por elementos de una condición.

Las Relaciones se expresan: $x R y$ “x tiene una relación con y”.

Las relaciones se pueden graficar en la matriz FAM.

A continuación un ejemplo de relación entre el conjunto M y N, donde: $m - n < 5$.

$$M = \{ 9, 10, 11, 12, 13, 14 \}$$

$$N = \{ 5, 6, 7, 8, 9, 10 \}$$

$$M * N = \left\{ \begin{array}{l} (9 \ 5) \ (9 \ 6) \ (9 \ 7) \ (9 \ 8) \ (9 \ 9) \ (9 \ 10) \\ (10 \ 5) \ (10 \ 6) \ (10 \ 7) \ (10 \ 8) \ (10 \ 9) \ (10 \ 10) \\ (11 \ 5) \ (11 \ 6) \ (11 \ 7) \ (11 \ 8) \ (11 \ 9) \ (11 \ 10) \\ (12 \ 5) \ (12 \ 6) \ (12 \ 7) \ (12 \ 8) \ (12 \ 9) \ (12 \ 10) \\ (13 \ 5) \ (13 \ 6) \ (13 \ 7) \ (13 \ 8) \ (13 \ 9) \ (13 \ 10) \\ (14 \ 5) \ (14 \ 6) \ (14 \ 7) \ (14 \ 8) \ (14 \ 9) \ (14 \ 10) \end{array} \right\}$$

Condición: $m - n < 5$

Relación: $m R n$

$$m R n = \left\{ \begin{array}{cccccc} (9 \quad 5) & (9 \quad 6) & (9 \quad 7) & (9 \quad 8) & (9 \quad 9) & (9 \quad 10) \\ F & (10 \quad 6) & (10 \quad 7) & (10 \quad 8) & (10 \quad 9) & (10 \quad 10) \\ F & F & (11 \quad 7) & (11 \quad 8) & (11 \quad 9) & (11 \quad 10) \\ F & F & F & (12 \quad 8) & (12 \quad 9) & (12 \quad 10) \\ F & F & F & F & (13 \quad 9) & (13 \quad 10) \\ F & F & F & F & F & (14 \quad 10) \end{array} \right\}$$

$\begin{array}{c} \text{N} \backslash \text{M} \\ \text{5} \end{array}$	9	10	11	12	13	14
5	1	1	1	1	1	1
6	0	1	1	1	1	1
7	0	0	1	1	1	1
8	0	0	0	1	1	1
9	0	0	0	0	1	1
10	0	0	0	0	0	1

Figura 1.10 FAM $m R n$.

1.2.4. REGLAS DE INFERENCIA.

En el bloque de Inferencia se deben seguir los siguientes pasos con el manejo de reglas:

- a) Para cada regla se define el grado de la premisa:

$$\mu_{a'}(x) = \mu_{a1}(x1) * \mu_{a2}(x2) * \mu_{a3}(x3) * \dots$$

- b) Para cada regla se infiere el conjunto de salida fuzzy:

$$\mu_{B'}(y) = \mu_{a(x)} + \mu_{B(y)}$$

- c) El conjunto fuzzy resultante para todas las reglas se obtiene combinando todos los conjuntos fuzzy de salida:

$$\mu_{B \text{ total}}(y) = \mu_{A^1(y)} \oplus \mu_{A^2(y)} \oplus \mu_{A^3(y)} \oplus \dots$$

Del conjunto de reglas que se deben generar para controlar todas las situaciones que se pueden presentar en el sistema, no siempre se activan todas, sólo se ejecutan las reglas que en ese momento se encuentren asociadas al problema.

Además se deben graficar por separado las reglas activadas con el mínimo, el máximo o el negativo valor de inferencia.

Se pueden realizar inferencias por el mínimo, inferencias por el máximo o inferencias negativas, todo depende de la regla, si la regla tiene como conector lógico AND entonces se realiza la inferencia por el mínimo, si la regla tiene como conector lógico OR entonces se realiza la Inferencia por el máximo, si la regla tiene como conector lógico NOT entonces se realiza la Inferencia negativa.

A continuación un ejemplo didáctico sin unidades de medida para simplificación.

Se va a fusificar la señal cuando el sensor de humedad registra el valor de 32 y el sensor de presión registra el valor de 24, la salida corresponde a la abertura de una válvula de un calefactor o de un actuador que regula las variables citadas.

La fusificación de la señal de entrada HUMEDAD sólo se realiza en los conjuntos difusos: Media y Alta, en la entrada PRESIÓN se realiza en los conjuntos difusos: Normal y Grande. Debido a que los valores de los sensores, influyen sólo en estos conjuntos fuzzy.

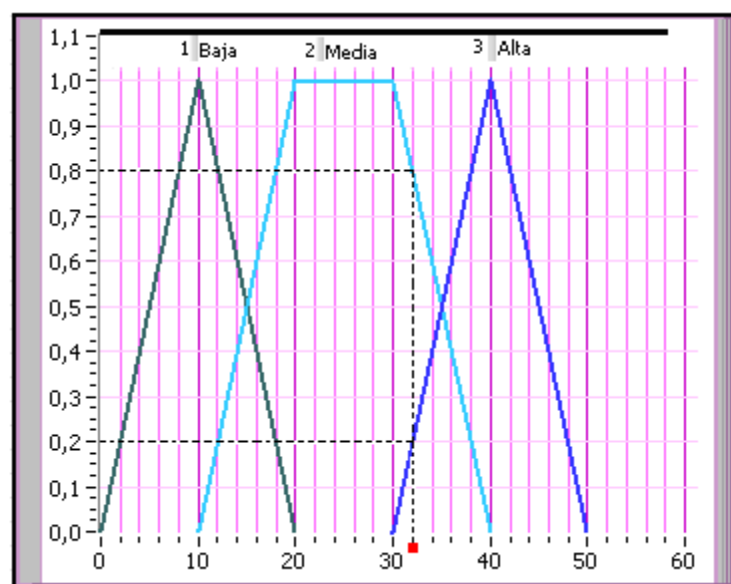


Figura 1.11 Entrada 1: HUMEDAD.

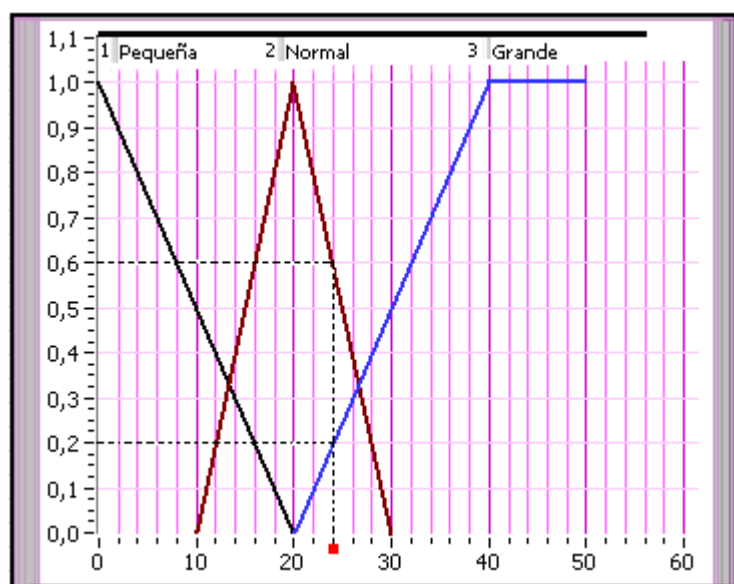


Figura 1.12 Entrada 2: PRESIÓN.

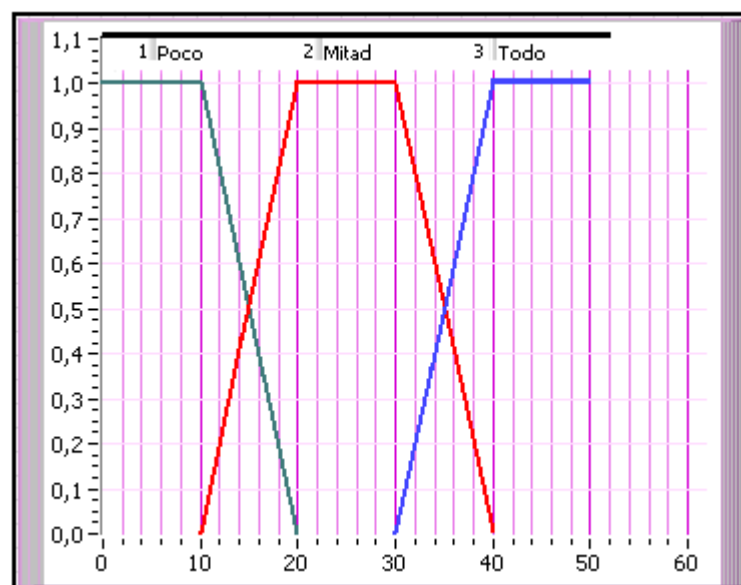


Figura 1.13 Salida: VÁLVULA.

Funciones de Membresía en entrada 1: HUMEDAD

$$\mu_{\text{Media}} = \begin{cases} x/10 - 1 & 10 \leq x \leq 20 \\ 1 & 20 \leq x \leq 30 \\ 4 - x/10 & 30 \leq x \leq 40 \end{cases}$$

$$\mu_{\text{Alta}} = \begin{cases} x/10 - 3 & 30 \leq x \leq 40 \\ 5 - x/10 & 30 \leq x \leq 40 \end{cases}$$

Funciones de Membresía en Entrada 2: PRESIÓN

$$\mu_{\text{Normal}} = \begin{cases} x/10 - 1 & 10 \leq x \leq 20 \\ 3 - x/10 & 20 \leq x \leq 30 \end{cases}$$

$$\mu_{\text{Grande}} = \begin{cases} x/20 - 1 & 20 \leq x \leq 40 \\ 1 & 40 \leq x \leq 50 \end{cases}$$

Membresías evaluadas en los valores de entrada:

$$\mu_{\text{Media}} = 4 - 32/10 = 0.8$$

$$\mu_{\text{Alta}} = 32/10 - 3 = 0.2$$

$$\mu_{\text{Normal}} = 3 - 24/10 = 0.6$$

$$\mu_{\text{Grande}} = 24/20 - 1 = 0.2$$

<i>salida:</i> VÁLVULA		HUMEDAD		
		Baja	Media	Alta
PRESIÓN	Pequeña	Poco	Poco	Mitad
	Normal	Poco	Mitad	Todo
	Grande	Mitad	Todo	Todo

Figura 1.14 FAM de ejemplo didáctico.

Reglas:

- IF Humedad = Baja AND Presión = Pequeña THEN Válvula = Poco.
 IF Humedad = Baja AND Presión = Normal THEN Válvula = Poco.
 IF Humedad = Baja AND Presión = Grande THEN Válvula = Mitad.
 IF Humedad = Media AND Presión = Pequeña THEN Válvula = Poco.
 IF Humedad = Media AND Presión = Normal THEN Válvula = Mitad.
 IF Humedad = Media AND Presión = Grande THEN Válvula = Todo.
 IF Humedad = Alta AND Presión = Pequeña THEN Válvula = Mitad.
 IF Humedad = Alta AND Presión = Normal THEN Válvula = Todo.
 IF Humedad = Alta AND Presión = Grande THEN Válvula = Todo.

Son 9 reglas en total pero sólo se activan las reglas de los valores de las entradas: Humedad Media y Humedad Alta con Presión Normal y Presión Grande.

Es decir sólo se ejecutan 4:

IF Humedad = Media AND Presión = Normal THEN Válvula = Mitad.

IF Humedad = Media AND Presión = Grande THEN Válvula = Todo.

IF Humedad = Alta AND Presión = Normal THEN Válvula = Todo.

IF Humedad = Alta AND Presión = Grande THEN Válvula = Todo.

Las reglas tienen el conector lógico AND por lo tanto se debe realizar la Inferencia por el mínimo:

$$R1: \mu \text{ Mitad} = \min(\mu \text{ Media}, \mu \text{ Normal}) = \min(0.8, 0.6) = 0.6$$

$$R2: \mu \text{ Todo} = \min(\mu \text{ Media}, \mu \text{ Grande}) = \min(0.8, 0.2) = 0.2$$

$$R3: \mu \text{ Todo} = \min(\mu \text{ Alta}, \mu \text{ Normal}) = \min(0.2, 0.6) = 0.2$$

$$R1: \mu \text{ Todo} = \min(\mu \text{ Alta}, \mu \text{ Grande}) = \min(0.2, 0.2) = 0.2$$

Generalmente cada regla tiene diferente valor y se grafican todas, por coincidencia en este ejemplo las 3 últimas reglas se repiten, por lo tanto esa regla sólo se grafica una vez:

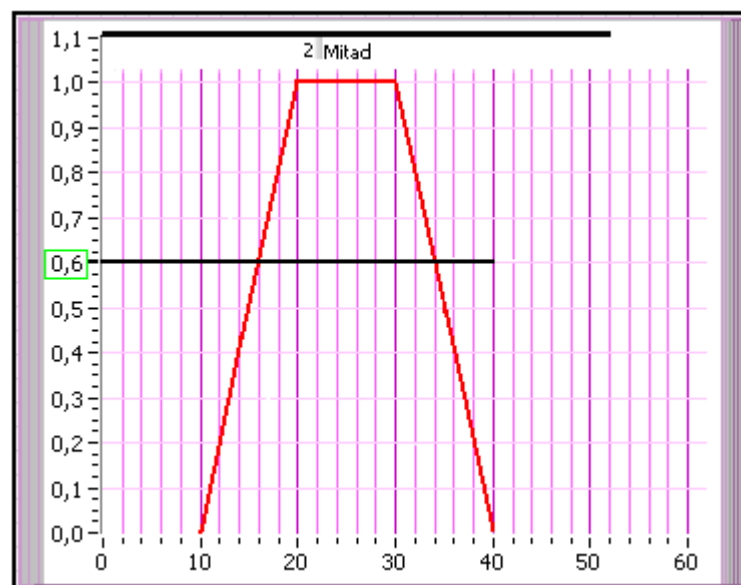


Figura 1.15 Gráfico de la Regla 1.

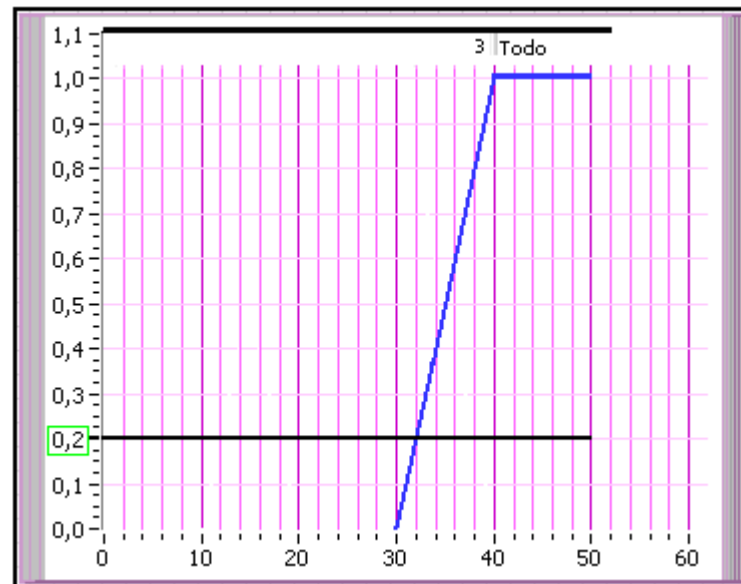


Figura 1.16 Gráfico de las Reglas 2, 3, 4.

Se debe realizar la Intersección entre el valor de Inferencia por el mínimo (mínimo, máximo o negativo) y el subconjunto difuso de salida, en ambos casos y luego la Unión entre las dos reglas.

Con la Intersección las gráficas son:

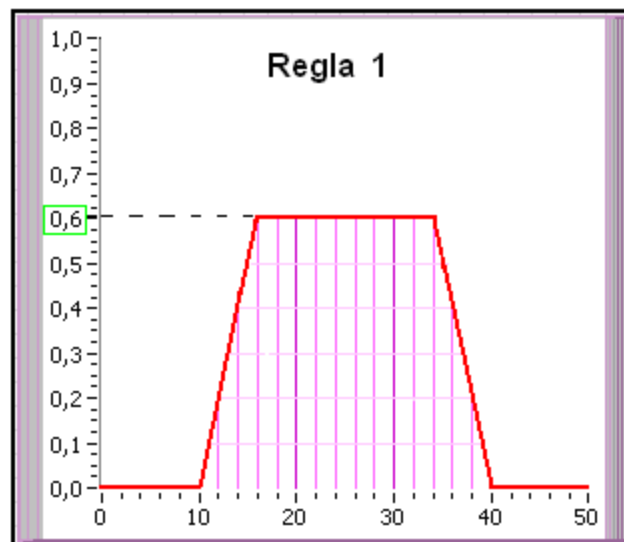


Figura 1.17 Intersección entre valor de inferencia y conjunto difuso Mitad.

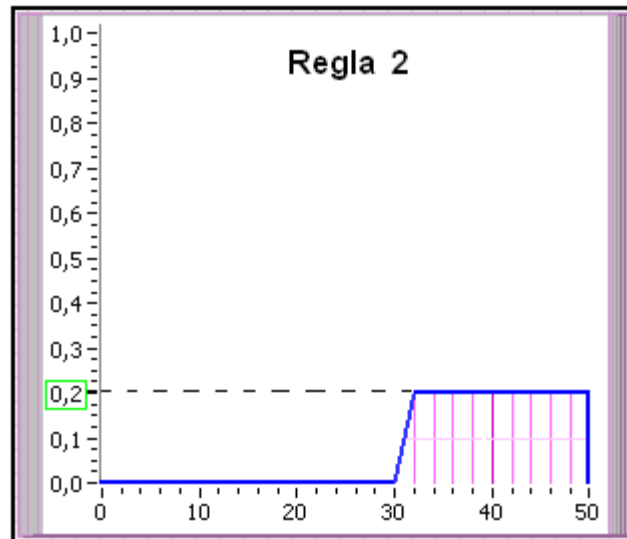


Figura 1.18 Intersección entre valor de inferencia y conjunto difuso Todo.

Para deducir los valores de la gráfica de Unión entre las dos reglas, se calculan los valores de x con la siguiente fórmula:

$$x = \frac{(x_2 - x_1)}{(y_2 - y_1)} * (y - y_1) + x_1$$

Para $0 \leq x \leq 10$ las dos reglas son 0, para $10 \leq x \leq 30$ el máximo es la regla 1, para los puntos críticos se calculan:

$$x = \frac{(40 - 30)}{(1 - 0)} * (0,2 - 0) + 30 \quad \rightarrow \quad x = 32 \quad \text{Regla 2 con 0,2}$$

$$x = \frac{(40 - 30)}{(0 - 1)} * (0,6 - 1) + 30 \quad \rightarrow \quad x = 34 \quad \text{Regla 1 con 0,6}$$

$$x = \frac{(40 - 30)}{(0 - 1)} * (0,2 - 1) + 30 \quad \rightarrow \quad x = 38 \quad \text{Regla 1 con 0,2}$$

Para $30 \leq x \leq 32$ el máximo es la regla 1, para $32 \leq x \leq 34$ el máximo es la regla 1, para $34 \leq x \leq 38$ el máximo es la regla 1, para $38 \leq x \leq 40$ el máximo es la regla 2, para $40 \leq x \leq 50$ el máximo es la regla 2.

Con estos valores ya se puede construir la gráfica final de la Unión entre las dos reglas, que también será la gráfica final para el bloque de desfusificación.

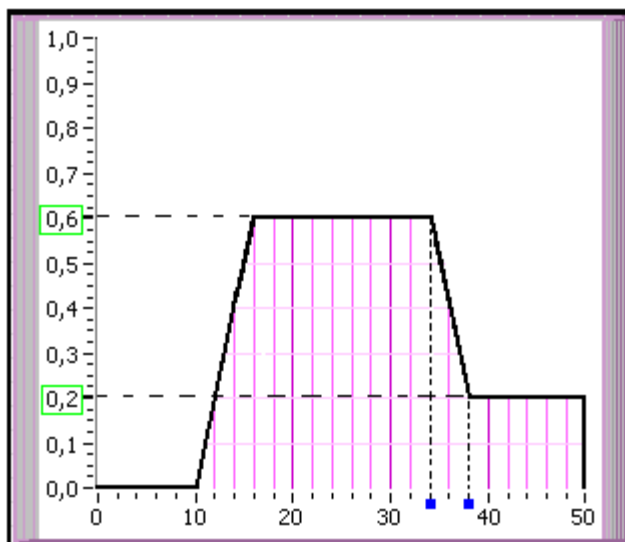


Figura 1.19 Gráfica Final (Unión de las reglas).

1.3. DESFUSIFICACIÓN.

El bloque de desfusificación realiza la función contraria al bloque de fusificación, el bloque fusificador tiene como entradas valores concretos de variables o lecturas de sensores y como salidas grados de pertenencia de conjuntos difusos entre 0 y 1 en cambio el bloque desfusificador tiene como entradas conjunto difusos de salida del bloque de inferencia y como salida valores concretos de variables que se asignarán a los actuadores. Para obtener, a partir del conjunto difuso de salida que resulta de la agregación de todas las reglas, una respuesta concreta o escalar, se deben aplicar métodos matemáticos.

Los 3 métodos más utilizados para resolver la desfusificación son:

- a) El método del punto máximo.
- b) El método del centro del área.
- c) El método del centro de gravedad.

1.3.1. PUNTO MÁXIMO.

Este método considera únicamente la acción de control sugerida por la regla cuya conclusión fue evaluada con el valor de verdad más alto. El punto de desfusificación encontrado por este método es el que se encuentra donde la salida difusa final alcanza su valor máximo por primera vez, en el sentido de los valores crecientes del eje horizontal.

En este método se elige como valor para la variable de salida la función de membresía máxima⁸. En general no es un método óptimo, ya que el valor máximo puede ser alcanzado por varias salidas:

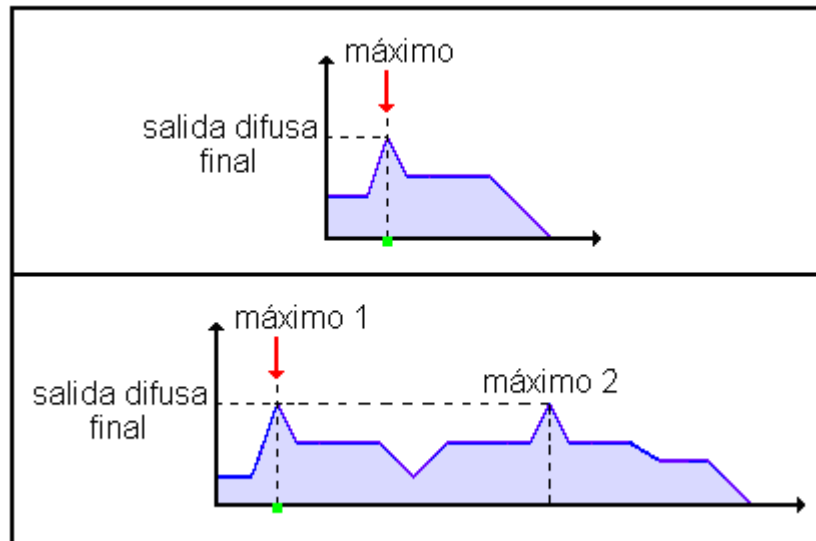


Figura 1.20 Punto máximo.

La fórmula es:

$$Z^* = z_j(o) \quad \text{Donde } z_j \text{ es la función de membresía máxima y } (o) \text{ cuando aparece por primera vez}$$

En el ejercicio anterior (Figura 1.19) el punto máximo es:

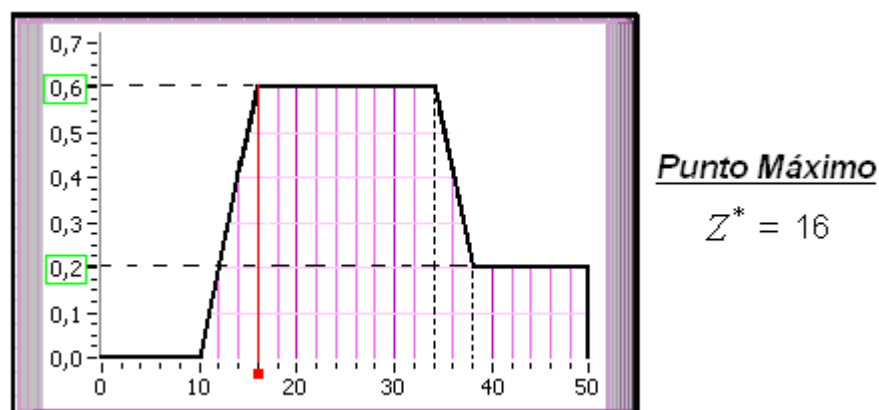


Figura 1.21 Ejemplo punto máximo.

Una mejora de éste método es la media del máximo⁹, que consiste en calcular el promedio entre los puntos máximos. Aquí el valor para llevar a cabo la acción de

⁸ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/met_com.html Punto máximo.

control se obtiene tomando el promedio de los valores de membresía máximos, es decir, este método genera como valor desfusificado la media de todos los valores que alcanzan el mismo máximo en la salida difusa final:

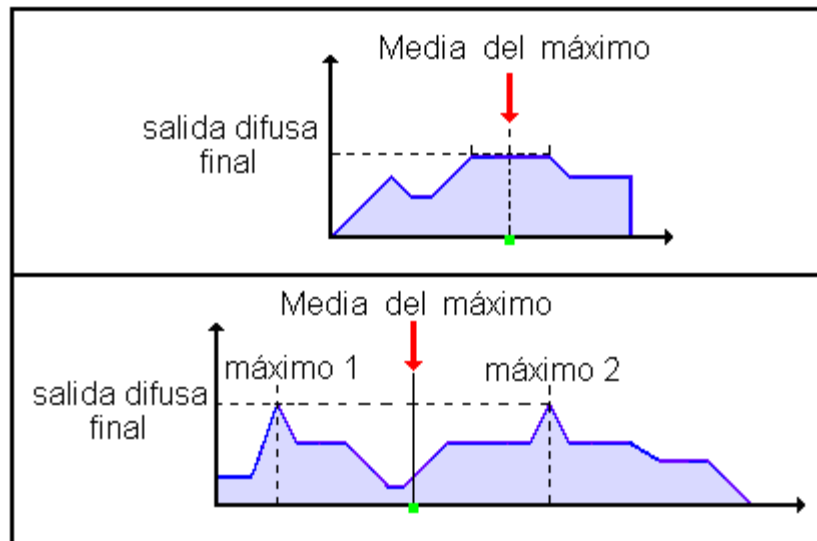


Figura 1.22 Media del Máximo.

La fórmula para el cálculo de la media del máximo es:

$$Z^* = \sum_{j=1}^l \frac{z_j}{l}$$

Donde l es el número de valores cuantizados z cuya función de membresía es máxima.

Estos métodos no son muy eficaces cuando las salidas difusas finales son bastante asimétricas, es decir cuando tienen formas muy irregulares como por ejemplo:

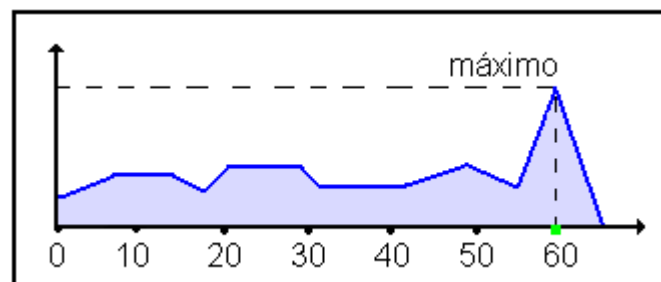


Figura 1.23 Máximo no representativo.

⁹ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/met_mom.html Mejora del punto máximo es la media del máximo que se obtiene calculando el promedio de todas las membresías iguales más altas.

En este caso el máximo se encuentra muy lejos y no representa un valor proporcional de defusificación, el valor debería estar entre 35 y 40 aproximadamente.

Para el ejercicio anterior (figura 1.19) el resultado de defusificación es:

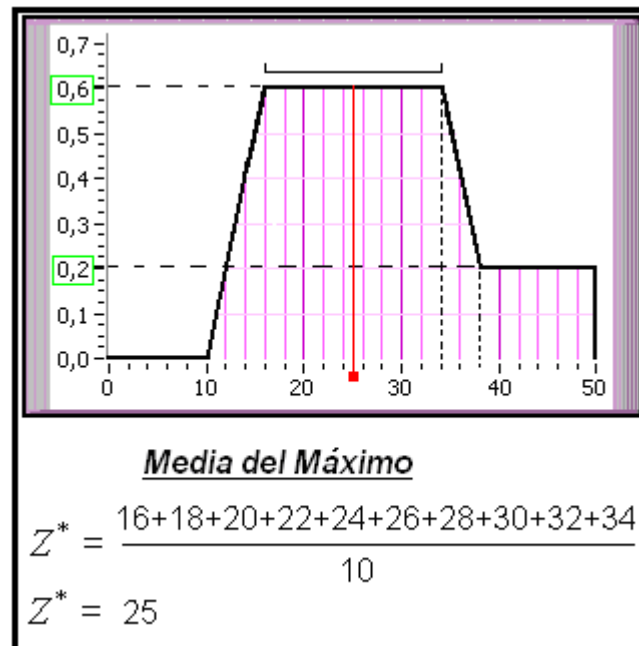


Figura 1.24 Ejemplo de media del máximo.

1.3.2. CENTRO DE ÁREA.

Este método es sensible al resultado de todas las reglas al momento de hacer la defusificación, lo que produce una salida de control más suave.

Asumimos que se produce una acción de control con una función de membresía dada por μ_c . Este método calcula el centro de área de la distribución para la acción de control. Asumiendo un universo de discurso discreto:¹⁰

$$Z^* = \frac{\sum_{j=1}^q z_j \mu_c(z_j)}{\sum_{j=1}^q \mu_c(z_j)}$$

Donde: q es el no de niveles de cuantización de salida,
 z_j es la suma de las salidas en el nivel de cuantización j
 $\mu_c(z_j)$ son los valores de la función de membresía en c .

En otras palabras, este método asigna el centro del área de la salida difusa final al valor defusificado. El centro de área también es llamado centroide.

¹⁰ Fórmula para calcular el centro del área: http://kim.ece.buap.mx/posgrado/Fuzzytutorial/met_coa.html

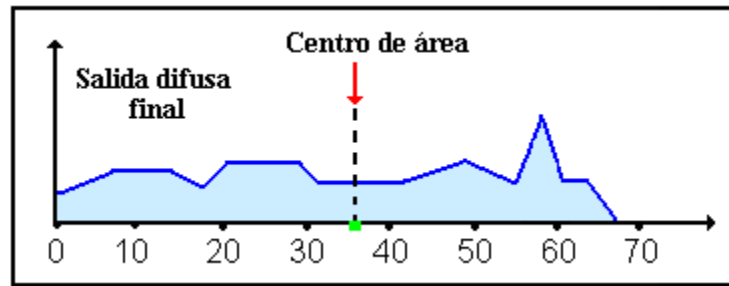


Figura 1.25 Centro de área.

El centro de área con la salida difusa final del ejemplo anterior (Figura 1.19) es:

Para facilitar el espacio físico, desglosamos la fórmula:

$$a = \sum_{j=1}^q Z_j \mu_c(Z_j) \quad b = \sum_{j=1}^q \mu_c(Z_j)$$

$$a = 10(0) + 12(0.2) + 14(0.4) + 16(0.6) + 18(0.6) + 20(0.6) + 22(0.6) + 24(0.6) + 26(0.6) + 28(0.6) + 30(0.6) + 32(0.6) + 34(0.6) + 36(0.4) + 38(0.2) + 40(0.2) + 42(0.2) + 44(0.2) + 46(0.2) + 48(0.2) + 50(0.2).$$

$$b = 0.2 + 0.4 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.6 + 0.4 + 0.2 + 0.2 + 0.2 + 0.2 + 0.2 + 0.2.$$

$$a = 218.4$$

$$b = 8.4$$

$$Z = a / b$$

$$Z = 218.4 / 8.4$$

$$Z = 26.$$

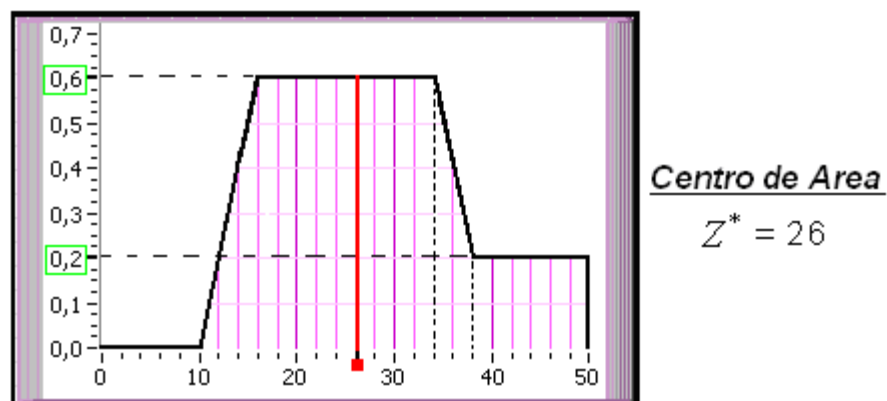


Figura 1.26 Ejemplo del centro de área.

1.3.3. CENTRO DE GRAVEDAD.

Utiliza como salida del sistema el centro de gravedad de la función característica de salida.

Es el método más utilizado en aplicaciones de la lógica difusa a la ingeniería, ya que se obtiene una solución única, aunque a veces es difícil de calcular.

La fórmula para el cálculo del centro de gravedad es:

$$Z^* = \frac{\int z \cdot \mu(z) \delta z}{\int \mu(z) \delta z} \quad \text{Donde } \mu(z) \text{ es la función de membresía.}$$

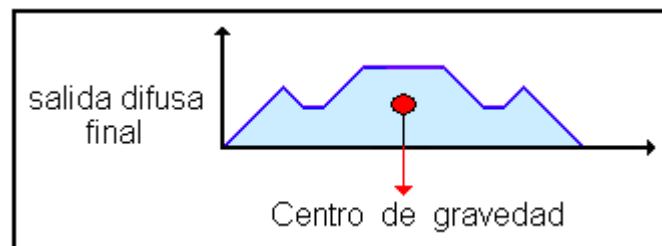


Figura 1.27 Centro de gravedad.

A continuación el centro de gravedad de la salida difusa final del ejercicio anterior (Figura 1.19). Para facilitar el espacio físico, desglosamos la fórmula:

$$a = \int z \cdot \mu(z) \delta z \quad b = \int \mu(z) \delta z$$

$$a = \int_{10}^{16} z (0.1z - 1) \delta z + \int_{16}^{34} z (0.6) \delta z + \int_{34}^{38} z (4 - 0.1z) \delta z + \int_{38}^{50} z (0.2) \delta z$$

$$a = \int_{10}^{16} 0.1z^2 - z \delta z + \int_{16}^{34} z (0.6) \delta z + \int_{34}^{38} 4z - 0.1z^2 \delta z + \int_{38}^{50} z (0.2) \delta z$$

$$a = \int_{10}^{16} 0.1z^2 \delta z - \int_{10}^{16} z \delta z + \int_{16}^{34} 0.6z \delta z + \int_{34}^{38} 4z \delta z - \int_{34}^{38} 0.1z^2 \delta z + \int_{38}^{50} 0.2z \delta z$$

$$a = \left[\frac{z^3}{30} \right]_{10}^{16} - \left[\frac{z^2}{2} \right]_{10}^{16} + \left[0.3z^2 \right]_{16}^{34} + \left[2z^2 \right]_{34}^{38} - \left[\frac{z^3}{30} \right]_{34}^{38} + \left[0.1z^2 \right]_{38}^{50}$$

$$a = \frac{2048}{15} - \frac{100}{3} - 128 + 50 + \frac{1734}{5} - \frac{384}{5} + 2888 - 2312 - \frac{27436}{15} + \frac{19652}{15} + 250 - \frac{722}{5}$$

$$a = \frac{6868}{15}$$

$$\begin{aligned}
 b &= \int_{10}^{16} (0.1z - 1) \delta z + \int_{16}^{34} (0.6) \delta z + \int_{34}^{38} (4 - 0.1z) \delta z + \int_{38}^{50} (0.2) \delta z \\
 b &= \int_{10}^{16} 0.1z \delta z - \int_{10}^{16} \delta z + \int_{16}^{34} 0.6 \delta z + \int_{34}^{38} 4 \delta z - \int_{34}^{38} 0.1z \delta z + \int_{38}^{50} 0.2 \delta z \\
 b &= \left[\frac{z^2}{20} \right]_{10}^{16} - \left[z \right]_{10}^{16} + \left[0.6z \right]_{16}^{34} + \left[4z \right]_{34}^{38} - \left[\frac{z^2}{20} \right]_{34}^{38} + \left[0.2z \right]_{38}^{50} \\
 b &= \frac{64}{5} - 5 - 16 + 10 + \frac{102}{5} - \frac{48}{5} + 152 - 136 - \frac{361}{5} + \frac{289}{5} + 10 - \frac{38}{5} \\
 b &= \frac{83}{5} \\
 Z^* &= \frac{a}{b} = \frac{\frac{6868}{15}}{\frac{83}{5}} = \frac{6868}{249} = 27.58
 \end{aligned}$$

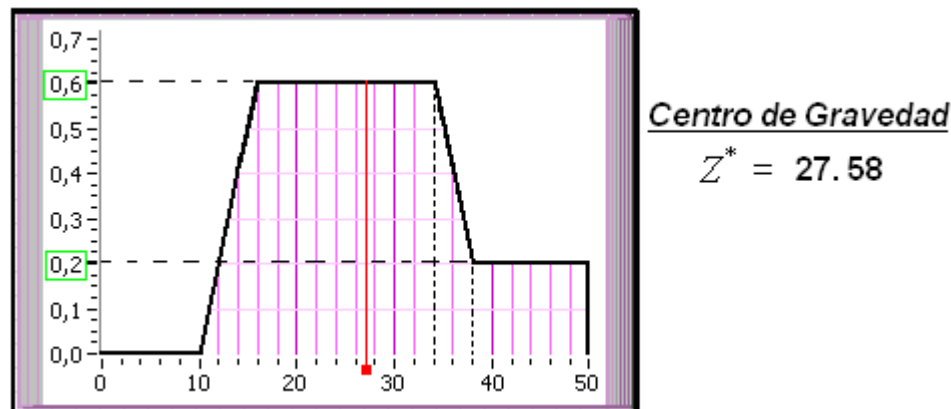


Figura 1.28 Ejemplo de centro de gravedad.

1.4. MODELOS DIFUSOS.

Los modelos difusos empleados en el control de procesos tienden a seguir la misma metodología empleada en el diseño de los sistemas de control clásicos¹¹, esto es: en primer lugar el diseño conceptual es hecho en papel una vez que se ha entendido tanto la mecánica del comportamiento del sistema como su dinámica en términos de entrada/salida.

¹¹ <http://kim.ece.buap.mx/posgrado/Fuzzytutorial/4intro.html> Modelos Difusos. Introducción.

Luego se procede a un ciclo de modelado, simulación y ajuste del sistema, y así sucesivamente se repite el ciclo hasta obtener el resultado deseado.

El proceso descrito anteriormente puede apoyarse en herramientas de cómputo especializadas.

Para los sistemas difusos, el método de diseño se efectúa de acuerdo con el siguiente ciclo:

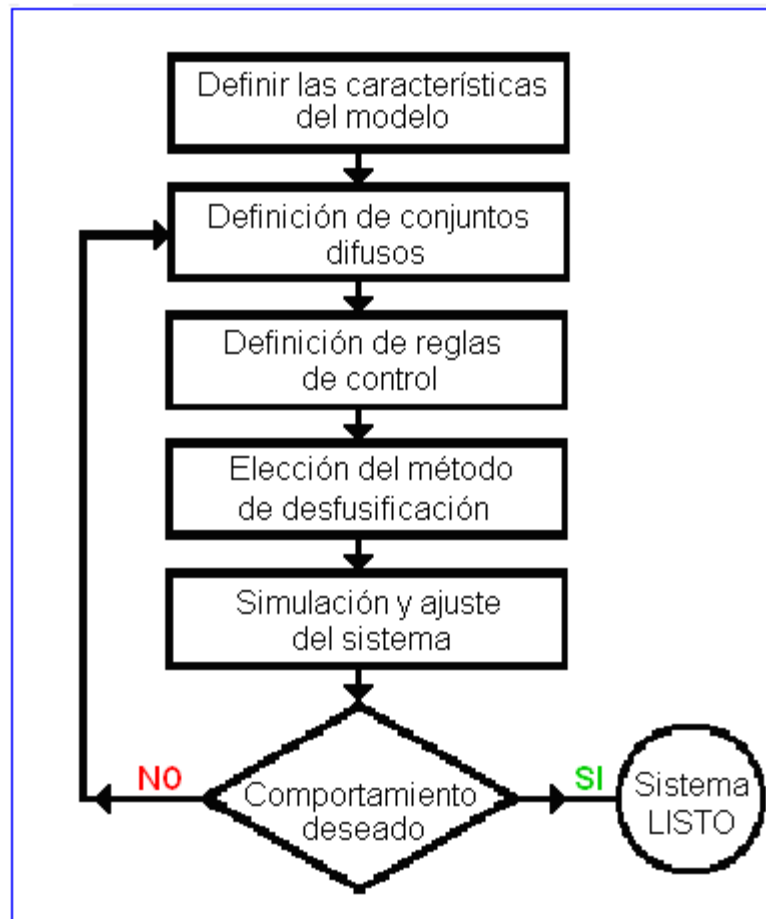


Figura 1.29 Metodología de Diseño.

A continuación se explican cada una de las etapas considerando el diagrama anterior.

1.4.1 DEFINICIÓN DE CARACTERÍSTICAS.

En esta etapa se definen las características funcionales y operacionales del modelo, aquí la tarea del diseñador consiste en definir aunque se carezca de un modelo matemático del sistema:

- a) Los datos de entrada al sistema.
- b) Las transformaciones básicas que se aplicarán a los datos.
- c) Los datos de salida del sistema.

También se debe definir donde exactamente influye el proceso difuso en la arquitectura total del sistema¹², con la finalidad de proporcionar una clara visión de la forma en que los datos estarán fluyendo hacia y desde el sistema difuso, además de que proporciona una gran ayuda al diseñador para la estimación de los números y rangos de las entradas y salidas difusas requeridas.

El presente trabajo de tesis consta de:

Definición de entradas:

Se ha establecido dos entradas: Temperatura y Radiación de Luz.

A la primera entrada le corresponden los valores que nos entregue el sensor de Temperatura y a la segunda le corresponden los valores que nos entregue el sensor de luz.

Definición de transformaciones básicas que se aplicarán a los datos:

Las transformaciones básicas de los datos se realizan debido a que los sensores y transductores reciben estimulación de temperatura o radiación de luz, en este caso, para entregar variaciones de voltaje o corriente a su salida, más no valores absolutos.

Entonces se debe primero calcular estas variaciones de voltaje para saber a que temperatura o radiación de luz corresponden.

Definición de salidas:

Se ha establecido dos salidas: Acción de respuesta a la temperatura (Calefactor / Ventilador) y Lámpara Fluorescente.

De igual forma a la primera salida le corresponden los valores que asignamos a los actuadores Calefactor y Ventilador y a la segunda salida le corresponden los valores que asignamos al actuador Lámpara.

¹² http://kim.ece.buap.mx/posgrado/Fuzzytutorial/carac_modelo.html Definición de características de los modelos difusos.

1.4.2. DEFINICIÓN DE CONJUNTOS DIFUSOS.

Para el proceso de definición de los conjuntos difusos en sistemas de control, primero se identifican y nombran las variables de entrada y de salida y se establecen sus rangos.

El proyecto contiene:

El sensor de Temperatura recibe estimulaciones en rangos bastante amplios desde -10°C hasta 150°C , como necesitamos crear un ambiente cómodo y además en nuestro entorno no se dan este tipo de valores en temperatura, consideramos importante reducir el rango de 0 a 50°C , por lo tanto ese será el conjunto universo de discursión para la primera entrada. El sensor de Radiación de Luz actúa en un rango de 0 a 10 voltios, este será el conjunto universo para la segunda entrada. Para las salidas, el conjunto U será del 0 al 100% de la acción de los actuadores.

ENTRADAS: Temperatura [0, 50] $^{\circ}\text{C}$, Radiación de Luz [0, 10] voltios.

SALIDAS: Calefactor / Ventilador [0, 100] %, Lámpara [0, 100] %.

Como segundo punto, cada variable es descompuesta en un grupo de términos difusos. Cada término representa un conjunto difuso en el universo de discurso de la variable.

Algunas recomendaciones para la definición de los conjuntos difusos:

- El número de términos difusos (conjuntos) asociados a cada variable debe ser, generalmente un número impar entre 5 y 9.
- Para producir una acción de control suave, cada conjunto debe traslaparse un poco sobre los conjuntos vecinos.¹³ El traslapamiento debe ser entre 10% y 50% del espacio ocupado por el conjunto vecino. La suma de las membresías de los puntos verticales del traslapamiento siempre debe ser igual o menor a 1.
- La densidad de los conjuntos difusos debe ser mayor alrededor del punto óptimo de control del sistema y menor conforme aumenta la distancia a ese punto.

¹³ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/def_fuzzysets.html Traslamiento de conjuntos difusos.

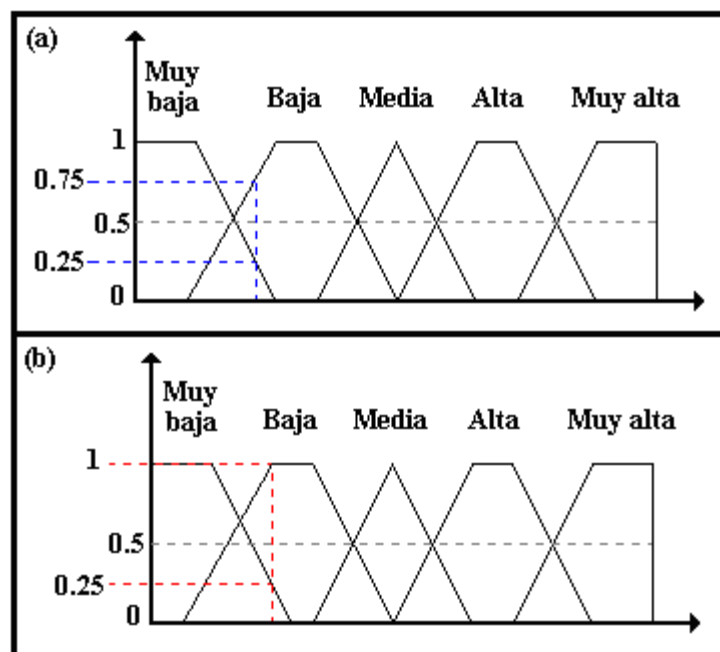


Figura 1.30 Traslapamiento de conjuntos difusos: a) Partición difusa traslapada correctamente. b) Partición difusa traslapada incorrectamente.

De acuerdo con las recomendaciones anteriores un ejemplo de conjuntos de entrada y salida es:

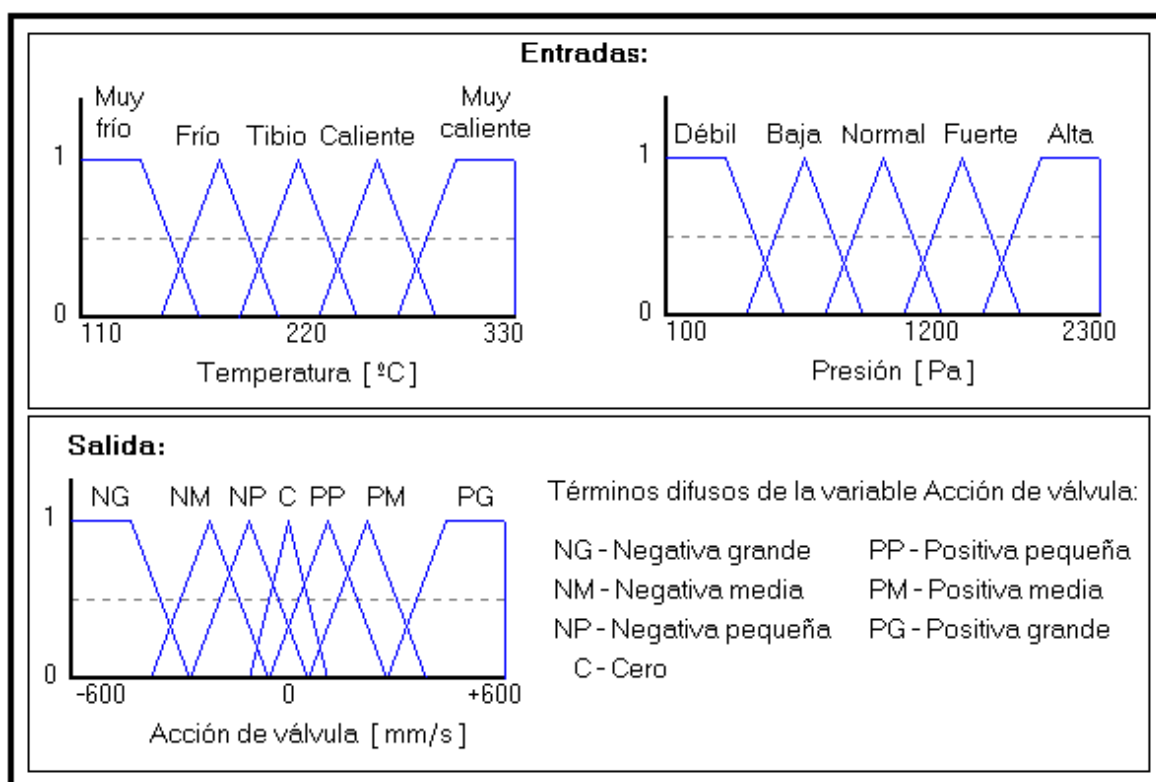


Figura 1.31 Ejemplo de conjuntos difusos de entrada y salida.

El proyecto de tesis presenta:

En la entrada Temperatura se tienen 9 conjuntos fuzzy denotados con las variables lingüísticas: Extra Frío, Muy Frío, Frío, Templado, Normal, Tibio, Cálido, Muy Cálido, Extra Caliente. En la entrada Radiación de Luz se tienen 5 conjuntos expresados con las variables lingüísticas: Débil, Leve, Moderado, Elevado, Fuerte.

En la salida Calefactor / Ventilador se tienen 9 conjuntos fuzzy: Calefactor Total (CTotal), Calefactor Suficiente (CSufi), Calefactor Normal (CNormal), Calefactor Mínimo (CMin), Calefactor y Ventilador Apagados (CAVA), Ventilador Mínimo (VMin), Ventilador Normal (VNormal), Ventilador Suficiente (VSufi), Ventilador Total (VTotal). En la salida Lámpara se tienen 5 conjuntos: Lámpara Apagada (LNada), Lámpara Baja (LBaja), Lámpara Media (LMedia), Lámpara Suficiente (LSufi), Lámpara Total (LTotal):

Cuando hablamos de conjunto universo de discurso, los conjuntos difusos pasan a ser subconjuntos del conjunto U .

A continuación se detallan las entradas y salidas del sistema fuzzy y además las funciones de membresía que corresponden a cada subconjunto difuso:

Temperatura:

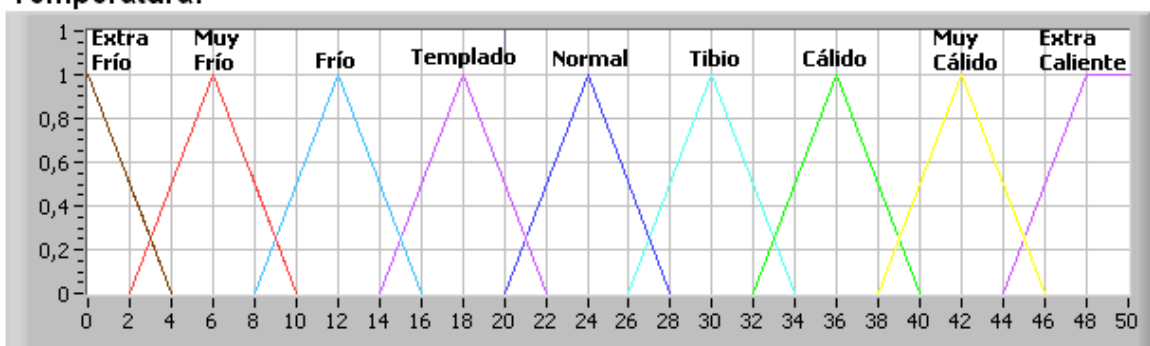


Figura 1.32 Entrada Temperatura.¹⁴

$$\mu_{\text{ExtraFrío}} = \{1 - x/4 \quad 0 \leq x \leq 4\}$$

¹⁴ Variable Lingüística: TEMPERATURA “conjunto universo”, Variables Lingüísticas: Extra Frío, Muy Frío, Frío, Templado, Normal, Tibio, Cálido, Muy Cálido y Extra Caliente “subconjuntos difusos”.

$$\mu_{\text{MuyFrío}} = \begin{cases} x/4 - 1/2 & 2 \leq x \leq 6 \\ 5/2 - x/4 & 6 \leq x \leq 10 \end{cases}$$

$$\mu_{\text{Frío}} = \begin{cases} x/4 - 2 & 8 \leq x \leq 12 \\ 4 - x/4 & 12 \leq x \leq 16 \end{cases}$$

$$\mu_{\text{Templado}} = \begin{cases} x/4 - 7/2 & 14 \leq x \leq 18 \\ 11/2 - x/4 & 18 \leq x \leq 22 \end{cases}$$

$$\mu_{\text{Normal}} = \begin{cases} x/4 - 5 & 20 \leq x \leq 24 \\ 7 - x/4 & 24 \leq x \leq 28 \end{cases}$$

$$\mu_{\text{Tibio}} = \begin{cases} x/4 - 13/2 & 26 \leq x \leq 30 \\ 17/2 - x/4 & 30 \leq x \leq 34 \end{cases}$$

$$\mu_{\text{Cálido}} = \begin{cases} x/4 - 8 & 32 \leq x \leq 36 \\ 10 - x/4 & 36 \leq x \leq 40 \end{cases}$$

$$\mu_{\text{MuyCálido}} = \begin{cases} x/4 - 19/2 & 38 \leq x \leq 42 \\ 23/2 - x/4 & 42 \leq x \leq 46 \end{cases}$$

$$\mu_{\text{ExtraCaliente}} = \begin{cases} x/4 - 11 & 44 \leq x \leq 48 \\ 1 & 48 \leq x \leq 50 \end{cases}$$

Radiación de Luz:

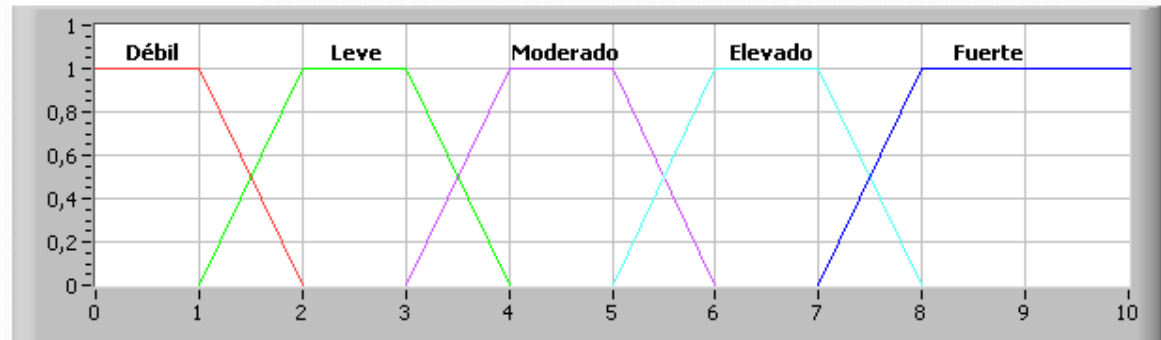


Figura 1.33 Entrada Radiación de Luz.

$$\mu_{\text{Débil}} = \begin{cases} 1 & 0 \leq x \leq 1 \\ 2 - x & 1 \leq x \leq 2 \end{cases}$$

$$\mu_{\text{Leve}} = \begin{cases} x - 1 & 1 \leq x \leq 2 \\ 1 & 2 \leq x \leq 3 \\ 4 - x & 3 \leq x \leq 4 \end{cases}$$

$$\mu_{\text{Moderado}} = \begin{cases} x - 3 & 3 \leq x \leq 4 \\ 1 & 4 \leq x \leq 5 \\ 6 - x & 5 \leq x \leq 6 \end{cases}$$

$$\mu_{\text{Elevado}} = \begin{cases} x - 5 & 5 \leq x \leq 6 \\ 1 & 6 \leq x \leq 7 \\ 8 - x & 7 \leq x \leq 8 \end{cases}$$

$$\mu_{\text{Fuerte}} = \begin{cases} x - 7 & 7 \leq x \leq 8 \\ 1 & 8 \leq x \leq 10 \end{cases}$$

Calefactor / Ventilador:

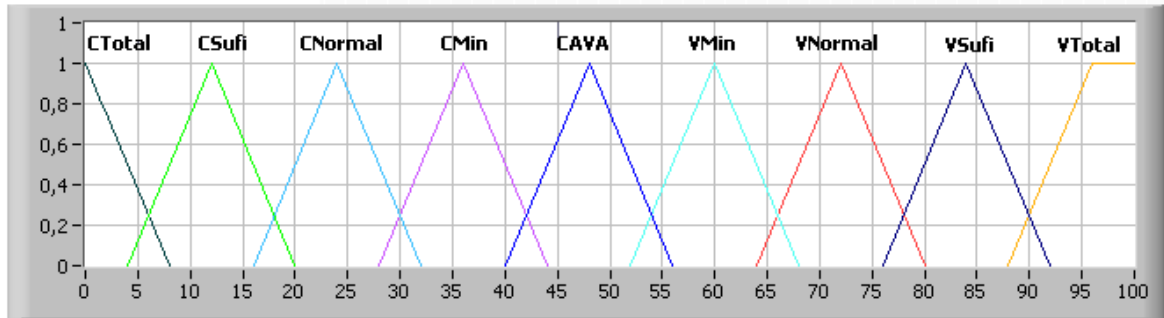


Figura 1.34 Salida Acción Respuesta Temperatura (Calefactor / Ventilador).

$$\mu_{\text{CTotal}} = \begin{cases} 1 - x/8 & 0 \leq x \leq 8 \end{cases}$$

$$\mu_{\text{CSufi}} = \begin{cases} x/8 - 1/2 & 4 \leq x \leq 12 \\ 5/2 - x/8 & 12 \leq x \leq 20 \end{cases}$$

$$\mu_{\text{CNormal}} = \begin{cases} x/8 - 2 & 16 \leq x \leq 24 \\ 4 - x/4 & 24 \leq x \leq 32 \end{cases}$$

$$\mu_{\text{CMin}} = \begin{cases} x/8 - 7/2 & 28 \leq x \leq 36 \\ 11/2 - x/8 & 36 \leq x \leq 44 \end{cases}$$

$$\mu_{\text{CAVA}} = \begin{cases} x/8 - 5 & 40 \leq x \leq 48 \\ 7 - x/8 & 48 \leq x \leq 56 \end{cases}$$

$$\mu_{\text{VMin}} = \begin{cases} x/8 - 13/2 & 52 \leq x \leq 60 \\ 7/2 - x/8 & 60 \leq x \leq 68 \end{cases}$$

$$\mu_{\text{VNormal}} = \begin{cases} x/8 - 8 & 64 \leq x \leq 72 \\ 10 - x/8 & 72 \leq x \leq 80 \end{cases}$$

$$\mu_{VSufi} = \begin{cases} x/8 - 19/2 & 76 \leq x \leq 84 \\ 23/2 - x/8 & 84 \leq x \leq 92 \end{cases}$$

$$\mu_{VTotal} = \begin{cases} x/8 - 11 & 88 \leq x \leq 96 \\ 1 & 96 \leq x \leq 100 \end{cases}$$

Lámpara:

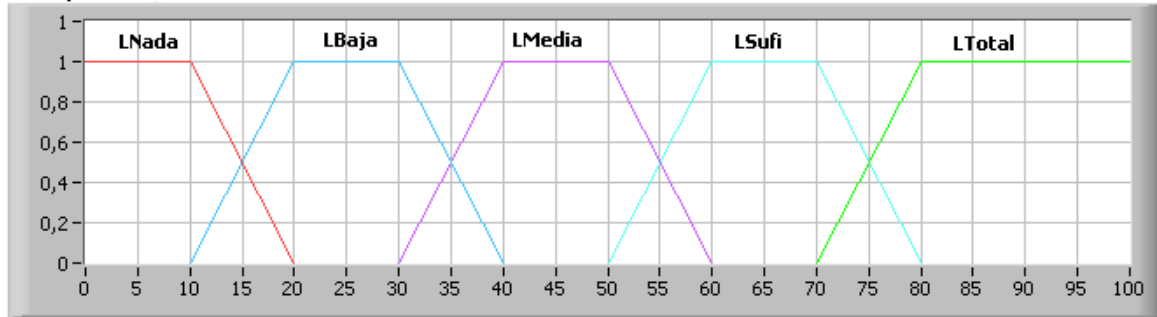


Figura 1.35 Salida Lámpara Fluorescente.

$$\mu_{LNada} = \begin{cases} 1 & 0 \leq x \leq 10 \\ 2 - x/10 & 10 \leq x \leq 20 \end{cases}$$

$$\mu_{LBaja} = \begin{cases} x/10 - 1 & 10 \leq x \leq 20 \\ 1 & 20 \leq x \leq 30 \\ 4 - x/10 & 30 \leq x \leq 40 \end{cases}$$

$$\mu_{LMedia} = \begin{cases} x/10 - 3 & 30 \leq x \leq 40 \\ 1 & 40 \leq x \leq 50 \\ 6 - x/10 & 50 \leq x \leq 60 \end{cases}$$

$$\mu_{LSufi} = \begin{cases} x/10 - 5 & 50 \leq x \leq 60 \\ 1 & 60 \leq x \leq 70 \\ 8 - x/10 & 70 \leq x \leq 80 \end{cases}$$

$$\mu_{LTotal} = \begin{cases} x/10 - 7 & 70 \leq x \leq 80 \\ 1 & 80 \leq x \leq 100 \end{cases}$$

1.4.3. DEFINICIÓN DE REGLAS.

Las reglas de control engloban el conocimiento del sistema y los objetivos de control. Cada regla tiene un estado del sistema en su premisa y una acción de control sugerida en su conclusión.¹⁵

¹⁵ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/def_reglas.html Definición de reglas de control.

Las reglas de control difusas conectan los valores de entrada con las propiedades de la salida del modelo. Están expresadas como proposiciones condicionales: SI [ESTADO DEL PROCESO] ENTONCES [ACCIÓN DE CONTROL].

Donde "Estado del proceso" y "Acción de control" es una proposición (o un grupo de proposiciones ligadas por un conectivo Y u O), de la forma:

v es T ; donde v es una variable y T un término difuso. Como ejemplo de regla de control tenemos:

SI Temperatura es Fría Y Presión es Alta ENTONCES Acción de válvula Es Positiva Media.

Las reglas de control difusas son declarativas y no secuenciales, lo que significa que el orden en que se expresan no es importante. Como una medida preventiva para el mantenimiento del controlador es recomendable agrupar las reglas de acuerdo a las variables de sus premisas.

El número de reglas que requiere un controlador difuso se halla multiplicando el número de términos difusos de las variables de entrada. De esta manera las reglas cubrirán todas las posibles combinaciones provenientes de las distintas entradas. Como ejemplo, para un sistema con dos variables de entrada cada una con 5 términos difusos existen $5 * 5 = 25$ combinaciones de entrada y por lo tanto 25 reglas de control.

1.4.3.1. Diseño de una base de reglas.

Para diseñar una base de reglas se utiliza una matriz FAM¹⁶ que cubre todas las posibles combinaciones de las entradas.

Para un sistema con dos entradas se asigna una entrada a cada eje de la matriz y existirán tantas divisiones en cada eje como términos difusos tenga la variable que le corresponde.¹⁷

En cada celda de la matriz se escribe la acción de control que sugeriría la regla que tendría esa combinación de entradas como premisa. De la matriz se pueden derivar todas las reglas que formarían una base completa.

¹⁶ FAM: Fuzzy Associative Memory que significa Memoria Asociativa Difusa.

¹⁷ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/dbase_reglas.html Diseño de una base de reglas.

Sí el sistema cuenta con tres entradas se utiliza una matriz por cada término lingüístico de la tercera variable.

En algunos casos es posible utilizar menos reglas pero no es recomendable hacerlo, puesto que las reglas representan conocimiento, si alguna es eliminada, se remueve conocimiento del sistema, conocimiento que puede volverse importante si el sistema es modificado después.

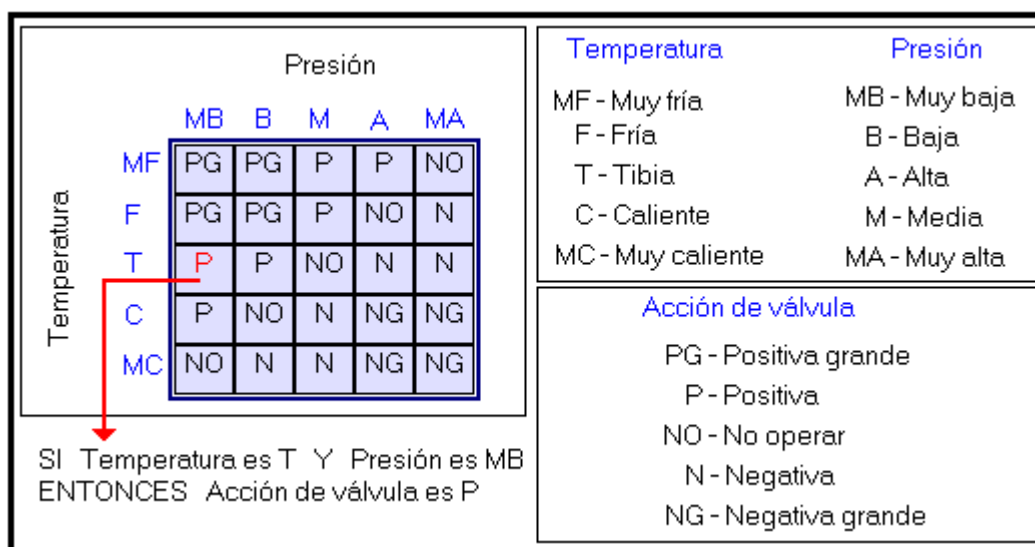


Figura 1.36 Ejemplo de una base de reglas.

El proyecto de tesis consta de las siguientes reglas de control:

IF	Temperatura es	Extra Frío	THEN	Calefactor/Ventilador es	CTotal
IF	Temperatura es	Muy Frío	THEN	Calefactor/Ventilador es	CSufi
IF	Temperatura es	Frío	THEN	Calefactor/Ventilador es	CNormal
IF	Temperatura es	Templado	THEN	Calefactor/Ventilador es	CMin
IF	Temperatura es	Normal	THEN	Calefactor/Ventilador es	CAVA
IF	Temperatura es	Tibio	THEN	Calefactor/Ventilador es	Vmin
IF	Temperatura es	Cálido	THEN	Calefactor/Ventilador es	VNormal
IF	Temperatura es	Muy Cálido	THEN	Calefactor/Ventilador es	VSufi
IF	Temperatura es	Extra Caliente	THEN	Calefactor/Ventilador es	VTotat
IF	Radiación Luz es	Débil	THEN	Lámpara es	LTotal
IF	Radiación Luz es	Leve	THEN	Lámpara es	LSufi
IF	Radiación Luz es	Moderado	THEN	Lámpara es	LMedia
IF	Radiación Luz es	Elevado	THEN	Lámpara es	LBaja
IF	Radiación Luz es	Fuerte	THEN	Lámpara es	LNada

La base de conocimientos consta de 9 reglas de control, aunque son 2 entradas, la primera con 9 variables y la segunda con 5 variables, debido a que la entrada Temperatura no tiene nada que ver con la entrada Radiación de luz. Cada entrada trabaja por separado y tiene su salida por separado.

	ENTRADA	SALIDA		
Temperatura	Extra Frío	CTotal	→	Calefactor Total
	Muy Frío	CSufi	→	Calefactor Suficiente
	Frío	CNormal	→	Calefactor Normal
	Templado	CMin	→	Calefactor Mínimo
	Normal	CAVA	→	Calefactor Apagado Ventilador Apagado
	Tibio	Vmin	→	Ventilador Mínimo
	Cálido	VNormal	→	Ventilador Normal
	Muy Cálido	VSufi	→	Ventilador Suficiente
	Extra Caliente	VTotat	→	Ventilador Total
Radiación Luz	Débil	LTotat	→	Lámpara Total
	Leve	LSufi	→	Lámpara Suficiente
	Moderado	LMedia	→	Lámpara Media
	Elevado	LBaja	→	Lámpara Baja
	Fuerte	LNada	→	Lámpara Nada

Figura 1.37 Base de reglas del proyecto.

1.4.4. ELECCIÓN DEL MÉTODO DE DESFUSIFICACIÓN.

Aquí se selecciona uno de los varios métodos para convertir un conjunto difuso de salida en una variable solución no difusa.¹⁸

Debido a que las reglas de control de la base de conocimientos, en el proyecto generan salidas difusas finales bastante regulares, se pueden elegir los métodos de punto máximo o media del máximo, pero estos métodos no son tan exactos, así que se selecciona el método del centro de área o centroide.

¹⁸ http://kim.ece.buap.mx/posgrado/Fuzzytutorial/detmeto_desdi.html
desfusificación.

Por ejemplo, si se fusificara la señal de la entrada temperatura en 14,2 y la entrada radiación de luz en 7,3 tendríamos una gráfica final de ésta forma:

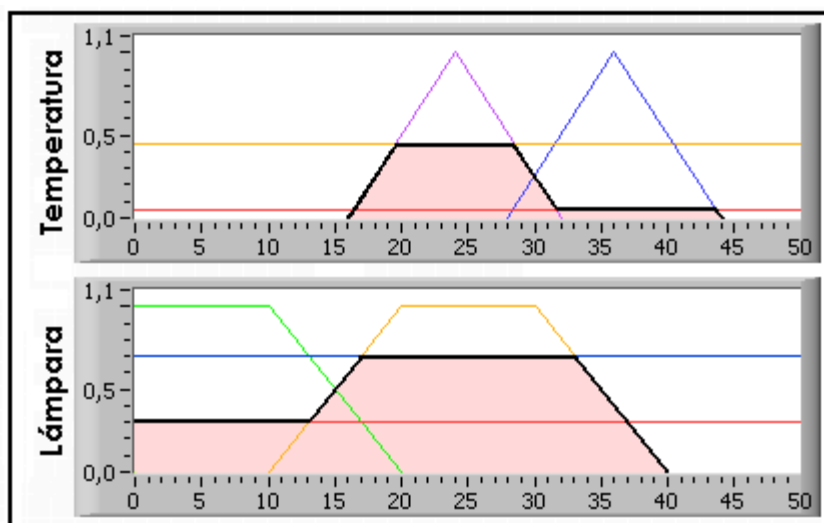


Figura 1.38 Salidas difusas finales del proyecto.

Además el método de desfusificación del Centro de área, es más fácil de calcular que el método de centro de gravedad, ya que se han establecido tanto en las entradas como en las salidas conjuntos difusos triangulares y trapezoidales, es decir conjuntos formados por rectas por lo que el cálculo de áreas es más fácil que el cálculo de integrales, que a su vez también resulta cálculo de áreas, pero con un procedimiento más complejo.

Si las salidas poseen conjuntos difusos sigmoidales o en campana de Gauss, el método más conveniente y que se acerca más a la realidad es el de centro de gravedad.

El centro de gravedad es el método más exacto y debe ser usado cuando el sistema necesita de mucha precisión en sus respuestas, en la presente tesis las respuestas serán asignadas a los actuadores: calefactor, ventilador y lámpara. El ser humano no percibe la diferencia si el ventilador por ejemplo gira al 100% de su velocidad o al 99,18%, pero si podría percibir si gira al 100% o al 90%.

En este caso el centro de gravedad es un gasto de cálculo matemático y computacional.

1.4.5. SIMULACIÓN Y AJUSTE DEL SISTEMA.

Cuando el modelo difuso ha sido construido, el proceso de simulación y desarrollo del prototipo comienza. En lo que se refiere a la simulación, esta puede realizarse en varios paquetes computacionales disponibles en el mercado. Generalmente estos paquetes incluyen herramientas para evaluar el modelo difuso y aislar problemas en los conjuntos difusos o en la base de reglas.¹⁹

Cuando los resultados de la simulación o las pruebas no son satisfactorios se realizan ajustes en las descripciones de los conjuntos difusos o en las acciones de control sugeridas por las reglas, hasta afinar el desempeño del control.

El modelo es comparado con casos similares conocidos para validar los resultados y terminar el desarrollo del control.

A continuación se realiza un ejemplo completo del proceso fuzzy en el proyecto de tesis. Se fusifica la señal en temperatura = 38,4 y en radiación de luz = 3,7:

Temperatura:

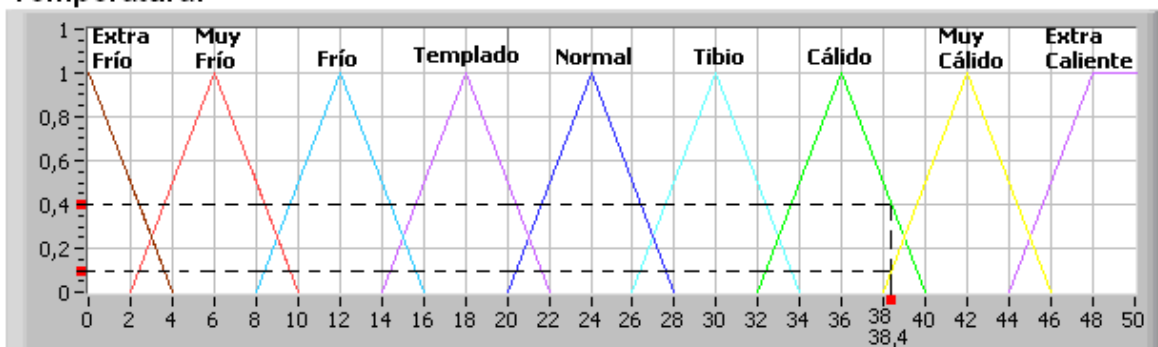


Figura 1.39 Entrada Temperatura.

Radiación de Luz:

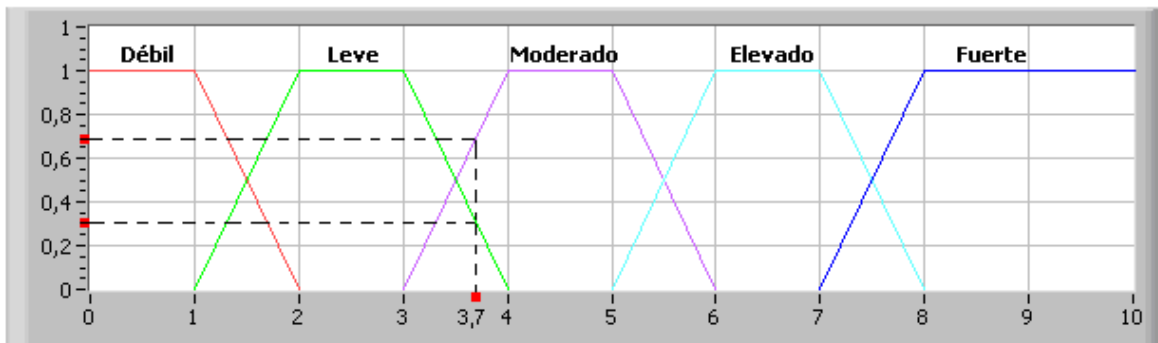


Figura 1.40 Entrada Radiación de Luz.

¹⁹ <http://kim.ece.buap.mx/posgrado/Fuzzytutorial/simyajuste.html> Simulación y ajuste del sistema.

Calefactor / Ventilador:

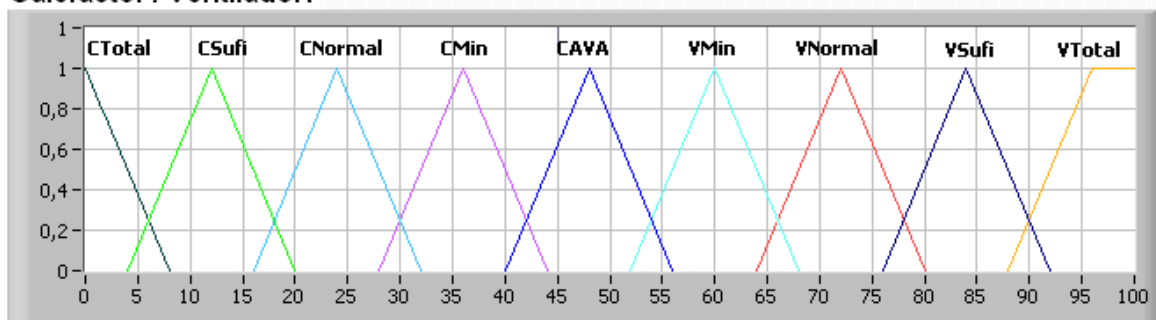


Figura 1.41 Salida Acción Respuesta Temperatura (Calefactor / Ventilador).

Lámpara:

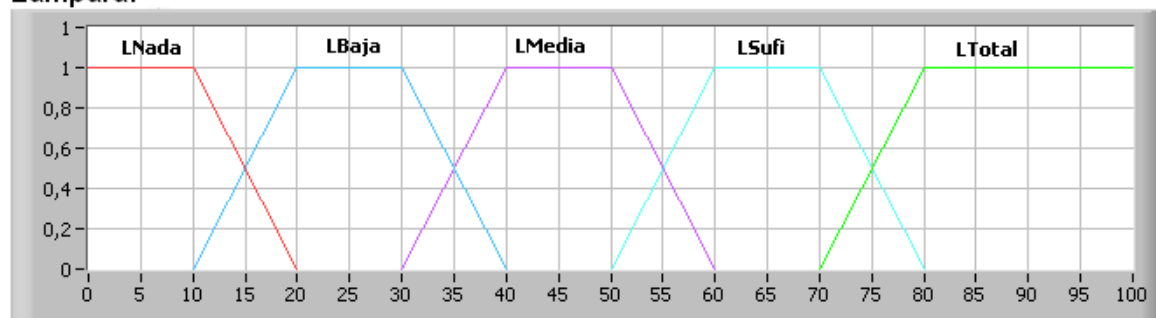


Figura 1.42 Salida Lámpara Fluorescente.

Las funciones de membresía de los conjuntos fuzzy asociados a los valores de la entrada Temperatura = 38,4

$$\mu_{\text{Cálido}} = \begin{cases} x/4 - 8 & 32 \leq x \leq 36 \\ 10 - x/4 & 36 \leq x \leq 40 \end{cases}$$

$$\mu_{\text{MuyCálido}} = \begin{cases} x/4 - 19/2 & 38 \leq x \leq 42 \\ 23/2 - x/4 & 42 \leq x \leq 46 \end{cases}$$

Las funciones de membresía de los conjuntos fuzzy asociados a los valores de la entrada Radiación de Luz = 3,7:

$$\mu_{\text{Leve}} = \begin{cases} x - 1 & 1 \leq x \leq 2 \\ 1 & 2 \leq x \leq 3 \\ 4 - x & 3 \leq x \leq 4 \end{cases}$$

$$\mu_{\text{Moderado}} = \begin{cases} x - 3 & 3 \leq x \leq 4 \\ 1 & 4 \leq x \leq 5 \\ 6 - x & 5 \leq x \leq 6 \end{cases}$$

$$\mu \text{ Cálido} = 10 - 38,4 / 4 = 0.4$$

$$\mu \text{ Muy Cálido} = 38,4 / 4 - 19 / 2 = 0.1$$

$$\mu \text{ Leve} = 4 - 3,7 = 0.3$$

$$\mu \text{ Moderado} = 3,7 - 3 = 0.7$$

Reglas:

- | | | | |
|---------|------------------------------|------|---------------------------------|
| 1.- IF | Temperatura = Extra Frío | THEN | Calefactor/Ventilador = CTotal |
| 2.- IF | Temperatura = Muy Frío | THEN | Calefactor/Ventilador = CSufi |
| 3.- IF | Temperatura = Frío | THEN | Calefactor/Ventilador = CNormal |
| 4.- IF | Temperatura = Templado | THEN | Calefactor/Ventilador = CMin |
| 5.- IF | Temperatura = Normal | THEN | Calefactor/Ventilador = CAVA |
| 6.- IF | Temperatura = Tibio | THEN | Calefactor/Ventilador = Vmin |
| 7.- IF | Temperatura = Cálido | THEN | Calefactor/Ventilador = VNormal |
| 8.- IF | Temperatura = Muy Cálido | THEN | Calefactor/Ventilador = VSufi |
| 9.- IF | Temperatura = Extra Caliente | THEN | Calefactor/Ventilador = VTotal |
| 10.- IF | Radiación Luz = Débil | THEN | Lámpara = LTotal |
| 11.- IF | Radiación Luz = Leve | THEN | Lámpara = LSufi |
| 12.- IF | Radiación Luz = Moderado | THEN | Lámpara = LMedia |
| 13.- IF | Radiación Luz = Elevado | THEN | Lámpara = LBaja |
| 14.- IF | Radiación Luz = Fuerte | THEN | Lámpara = LNada |

De todas las reglas sólo se activan:

- | | | | |
|---------|--------------------------|------|---------------------------------|
| 7.- IF | Temperatura = Cálido | THEN | Calefactor/Ventilador = VNormal |
| 8.- IF | Temperatura = Muy Cálido | THEN | Calefactor/Ventilador = VSufi |
| 11.- IF | Radiación Luz = Leve | THEN | Lámpara = LSufi |
| 12.- IF | Radiación Luz = Moderado | THEN | Lámpara = LMedia |

Regla 7 : VNormal “Ventilador Normal” → (0.4)

Regla 8 : VSufi “Ventilador Suficiente” → (0.1)

Regla 11 : LSufi “Lámpara Suficiente” → (0.3)

Regla 12 : LMedia “Lámpara Media” → (0.7)

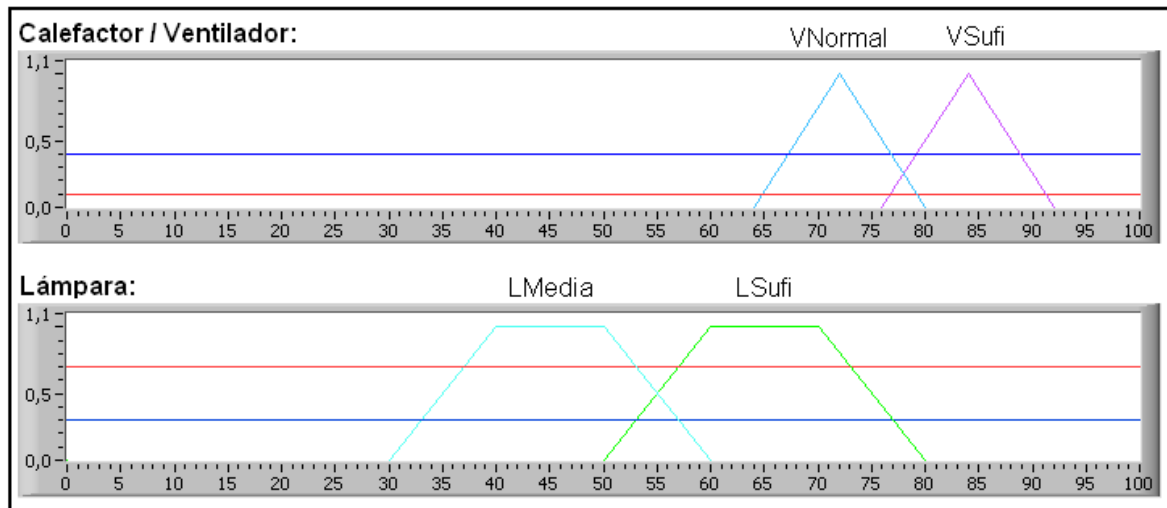


Figura 1.43 Conjuntos fuzzy y valores de inferencia.

Gráfico final: Intersección de Valores de Inferencia con conjuntos fuzzy y Unión entre reglas:

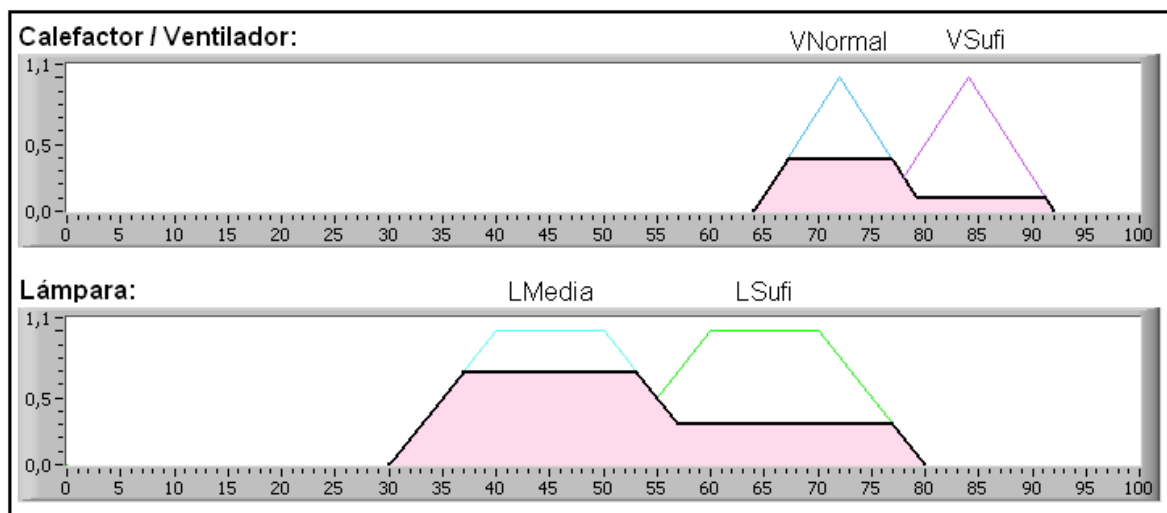


Figura 1.44 Grafica Final para la defusificación.

La defusificación se realiza por el método del centro de área.

Para Calefactor / Ventilador:

$$\mu_{VNormal} = \begin{cases} x/8 - 8 & 64 \leq x \leq 72 \\ 10 - x/8 & 72 \leq x \leq 80 \end{cases}$$

$$\mu_{VSufi} = \begin{cases} x/8 - 19/2 & 76 \leq x \leq 84 \\ 23/2 - x/8 & 84 \leq x \leq 92 \end{cases}$$

Punto 64: $64 / 8 - 8 = 0$

Punto 65: $65 / 8 - 8 = 0.125$

Punto 66: $66 / 8 - 8 = 0.25$

Punto 67: $67 / 8 - 8 = 0.375$

(0.4) $\rightarrow 8y + 64 = 8 * 0.4 + 64 = 67.2$

Puntos desde 68 hasta 76 = 0.4

(0.4) $\rightarrow 80 - 8y = 80 - 8 * 0.4 = 76.8$

Punto 77: $10 - 77 / 8 = 0.375$

Punto 78: $10 - 78 / 8 = 0.25$

Punto 79: $10 - 79 / 8 = 0.125$

(0.1) $\rightarrow 80 - 8y = 80 - 8 * 0.1 = 79.2$

Puntos desde 80 hasta 91 = 0.1

(0.1) $\rightarrow 92 - 8y = 92 - 8 * 0.1 = 91.2$

Punto 92: $23 / 2 - 92 / 8 = 0$

Fórmula:

$$Z^* = \frac{\sum_{j=1}^q Z_j \mu_c(Z_j)}{\sum_{j=1}^q \mu_c(Z_j)}$$

Desglosamos la fórmula:

$$a = \sum_{j=1}^q Z_j \mu_c(Z_j) \quad b = \sum_{j=1}^q \mu_c(Z_j)$$

$$\begin{aligned} a = & 64(0) + 65(0.125) + 66(0.25) + 67(0.375) + 68(0.4) + 69(0.4) + 70(0.4) \\ & + 71(0.4) + 72(0.4) + 73(0.4) + 74(0.4) + 75(0.4) + 76(0.4) + 77(0.375) + 78(0.25) + \\ & 79(0.125) + 80(0.1) + 81(0.1) + 82(0.1) + 83(0.1) + 84(0.1) + 85(0.1) + 86(0.1) \\ & + 87(0.1) + 88(0.1) + 89(0.1) + 90(0.1) + 91(0.1) + 92(0) \end{aligned}$$

$$a = 469.8$$

Puntos desde 57 hasta 76: 0.3

$$(0.3) \rightarrow 80 - 10 y = 80 - 10 * 0.3 = 77$$

Punto 77: $8 - 77 / 10 = 0.3$

Punto 78: $8 - 78 / 10 = 0.2$

Punto 79: $8 - 79 / 10 = 0.1$

Punto 80: $8 - 80 / 10 = 0$

Fórmula:

$$Z^* = \frac{\sum_{j=1}^q Z_j \mu_c(Z_j)}{\sum_{j=1}^q \mu_c(Z_j)}$$

Desglosamos la fórmula:

$$a = \sum_{j=1}^q Z_j \mu_c(Z_j) \quad b = \sum_{j=1}^q \mu_c(Z_j)$$

$$\begin{aligned} a = & 30(0) + 31(0.1) + 32(0.2) + 33(0.3) + 34(0.4) + 35(0.5) + 36(0.6) + 37(0.7) + \\ & 38(0.7) + 39(0.7) + 40(0.7) + 41(0.7) + 42(0.7) + 43(0.7) + 44(0.7) + 45(0.7) + \\ & 46(0.7) + 47(0.7) + 48(0.7) + 49(0.7) + 50(0.7) + 51(0.7) + 52(0.7) + 53(0.7) + \\ & 54(0.6) + 55(0.5) + 56(0.4) + 57(0.3) + 58(0.3) + 59(0.3) + 60(0.3) + 61(0.3) + \\ & 62(0.3) + 63(0.3) + 64(0.3) + 65(0.3) + 66(0.3) + 67(0.3) + 68(0.3) + 69(0.3) + \\ & 70(0.3) + 71(0.3) + 72(0.3) + 73(0.3) + 74(0.3) + 75(0.3) + 76(0.3) + 77(0.3) + \\ & 78(0.2) + 79(0.1) + 80(0) \end{aligned}$$

$$a = 1135.5$$

$$\begin{aligned} b = & 0 + 0.1 + 0.2 + 0.3 + 0.4 + 0.5 + 0.6 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + \\ & 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.7 + 0.6 + 0.5 + 0.4 + 0.3 + \\ & 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + \\ & 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.3 + 0.2 + 0.1 \end{aligned}$$

$$b = 22.1$$

$$Z = a / b$$

$$Z = 1135.5 / 22.1$$

$$Z = 51.38$$

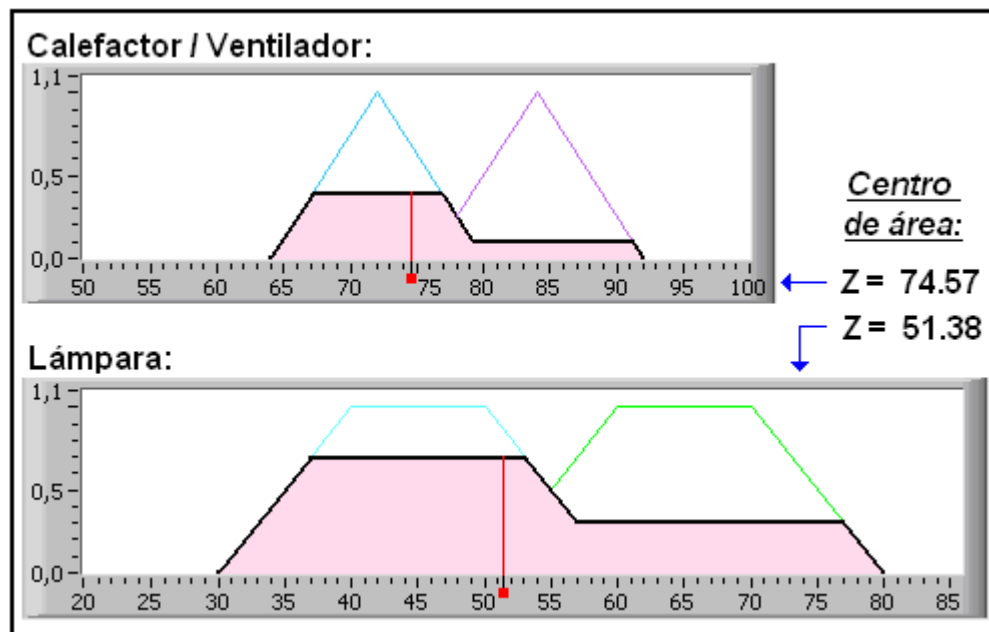


Figura 1.45 Valores de Defusificación.

El comportamiento del proceso difuso es conveniente, por lo tanto el sistema está listo.

1.5. APLICACIONES DE LÓGICA DIFUSA.

Algunas aplicaciones de lógica difusa que se desarrollan en varios campos, en la actualidad son:

- Diseño de un sistema de control difuso para una maqueta tipo invernadero.²⁰
- Aplicación de lógica difusa en el control de temperatura y monitoreo de señales de un intercambiador de calor.
- Diseño e implementación de una red Neuronal para la inserción de reglas en controladores difusos.

²⁰

http://scienti.colciencias.gov.co:8081/ciencia.war/search/EnGrupoInvestigacion/xmlInfo.do?nro_id_grupo=0000000000227#proyectos Algunos temas citados se encuentran en la raíz de esta página.

- Diseño e implementación de una herramienta software para desarrollo de memorias asociativas difusas FAM.²¹
- Diseño e implementación de software para el control neurodifuso de un modelo de grúa pórtico.
- Diseño e implementación de software para consultas difusas en bases de datos.
- Diseño e implementación de software para control de brazo mecánico con dos grados de libertad empleando lógica difusa.
- Software para clasificación de elementos mediante lógica difusa.
- Software para modelación de sistemas complejos mediante mapas cognitivos difusos.
- Software para el diseño, análisis, simulación e implementación de sistemas de lógica difusa.
- Herramienta de software basado en lógica difusa para toma de decisiones en rehabilitación por pérdida de miembros inferiores.
- Nuevas alternativas de secado mecánico de granos. Secado en lecho fluidizado utilizando sistemas de lógica difusa.
- Diseño de un modelo de toma de decisiones basadas en técnicas difusas.
- Hardware de control difuso basado en microcontrolador; programable desde Xfuzzy.
- Programación lineal difusa aplicada al despacho económico de carga.
- Modelo de lógica difusa y conjuntos difusos para el pronóstico de los niveles medios diarios del río Magdalena, en la estación limnigráfica de Puerto Salgar, Colombia.
- Diseño e Implementación del Control de Desplazamiento de un Robot Móvil Mediante Lógica Difusa.²²
- Lógica difusa aplicada a la selección de recurso humano.
- Lógica difusa aplicada a los sistemas expertos.²³
- Verificadores de ortografía, los cuales sugieren una lista de palabras probables para reemplazar una palabra mal escrita.
- Sistema Difuso para calefactor y aire acondicionado.

²¹ FAM Fuzzy Associative Memory (Memoria Asociativa Difusa).

²² <http://ozkr.net/escuela/maestria/fuzzy/Presentacion.pdf> Robot móvil con lógica difusa.

²³ <http://personales.ya.com/casanchi/mat/difusa01.htm#03> Aplicaciones de lógica difusa.

- Sistema experto basado en lógica difusa para diagnóstico de enfermedades (diabetes) en Prolog o Ehsis.
- Control de nivel con lógica difusa y labview.
- Lavadora de ropa, lava platos, que trabajan con lógica difusa.
- Sistema para determinar la calidad de alimentos por ejemplo arroz, mediante lógica difusa.
- Humidificadores y Deshumidificadores. Aumentar o reducir humedad en habitaciones, mediante lógica fuzzy.
- Mediante lógica difusa y el estudio de los genes se predice enfermedades como el cáncer.²⁴
- Localizador de luz infrarroja controlado con lógica difusa.
- Detección y clasificación automática de patrones en polisomnogramas de lactantes y niños aplicando lógica difusa y técnicas neuro-difusas.²⁵
- Evaluación de la aplicación de técnicas de lógica difusa en el control de máquinas eléctricas.
- Lógica difusa como herramienta para la bio indicación de la calidad del agua con macro invertebrados acuáticos en la Sábana de Bogotá – Colombia.
- Implementación de un circuito para compresión de imágenes aplicando lógica difusa.
- Control de semáforos utilizando lógica difusa.
- Control de un horno de cemento.
- Estabilización de imágenes en cámaras de video.
- Conducción automática de trenes metropolitanos subterráneos.
- Control de variables como: Presión, Temperatura, Humedad, Luminosidad, Nivel, Velocidad, Aceleración, Caudal, Voltaje, etc.

²⁴ http://www.lne.es/secciones/noticia.jsp?pRef=1697_38_539227_Cuencas-logica-difusa-ayudara-predecir-enfermedades-mediante-estudio-genes Estudio de genes y lógica difusa para el cáncer.

²⁵ <http://www.conicyt.cl/bases/fondecyt/proyectos/01/2003/1030545.html> polisomnogramas de lactantes y niños aplicando lógica difusa.

Gracias a que la lógica difusa se enfrenta con éxito a situaciones del mundo real, ha encontrado aplicaciones en una gran variedad de campos, de las cuales las más trascendentales se han dado en el área de control con el diseño e implementación de controladores difusos (Fuzzy Logic Control), iniciado por los trabajos de Mamdani y Assilian en los años setenta.²⁶

La lógica difusa tiene la habilidad de proporcionar un control inteligente en aplicaciones difíciles, especialmente aquellas que requieren de la optimización de muchas variables o el control de sistemas no lineales difíciles de modelar.

²⁶ <http://kim.ece.buap.mx/posgrado/Fuzzytutorial/aplicaciones.html> Aplicaciones de lógica difusa.

CAPÍTULO 2. SISTEMAS HMI / SCADA.

Las siglas SCADA significan: Supervisory Control And Data Acquisition, es decir: Adquisición, Control y Supervisión de Datos.

Las siglas HMI significan: Human Machine Interface, en español: Interfaz Hombre – Máquina.

Un sistema Scada incluye hardware y software. Se trata de una aplicación software especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo como controladores autónomos o autómatas programables y controlando el proceso de forma automática desde la pantalla del ordenador. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa: control de calidad, supervisión, mantenimiento, etc.

En este tipo de sistemas usualmente existe un ordenador, que efectúa tareas de supervisión y gestión de alarmas, así como tratamiento de datos y control de procesos. La comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.

Los programas necesarios y en su caso el hardware adicional que se necesite, se denomina: sistema SCADA²⁷.

Un Scada comprende todas aquellas soluciones de aplicación para referirse a la captura de información de un proceso o planta industrial (aunque no es absolutamente necesario que pertenezca a este ámbito), para que con esta información, sea posible realizar una serie de análisis o estudios con los que se pueden obtener valiosos indicadores que permitan una retroalimentación sobre un operador o sobre el propio proceso.

A continuación un gráfico de ejemplo sobre un sistema Scada aplicado en el área industrial:

²⁷ <http://www.automatas.org/redes/scadas.htm> Concepto de un sistema Scada.

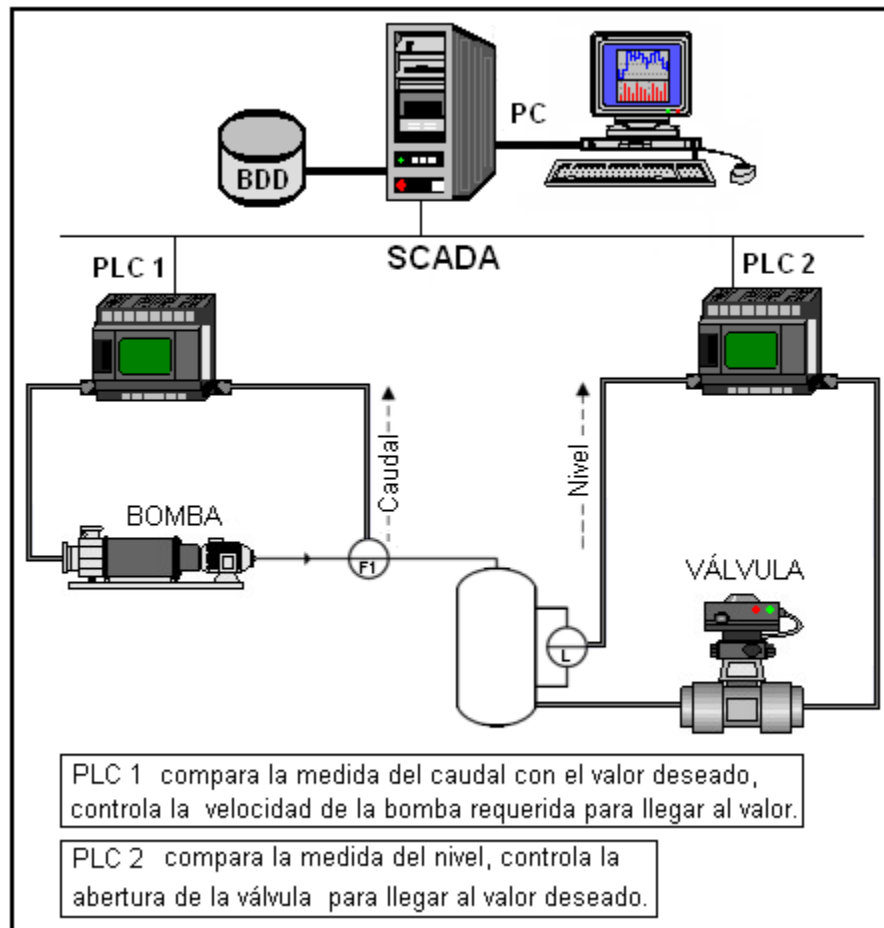


Figura 2.1 Estructura general de un sistema Scada.

Un sistema Scada incluye un hardware de señal de entrada y salida, controladores, interfaz hombre-máquina, redes, comunicaciones, base de datos y software.

El termino Scada usualmente se refiere a un sistema central que monitorea y controla un sitio completo o un sistema que se extiende sobre una gran distancia (kilómetros / millas). La mayor parte del control del sitio es en realidad realizada automáticamente por una Unidad Terminal Remota (RTU) o por un Controlador Lógico Programable (PLC).

Las funciones de control del servidor están casi siempre restringidas a reajustes básicos del sitio o capacidades de nivel de supervisión. Por ejemplo un PLC puede controlar el flujo de agua fría a través de un proceso, pero un scada puede permitirle a un operador cambiar el punto de consigna "set point" de control para el flujo y permitirá grabar y mostrar cualquier condición de alarma como la pérdida de un flujo o una alta temperatura.

La realimentación del lazo de control es cerrada a través del RTU o el PLC; el sistema Scada monitorea el desempeño general de dicho lazo.

Por otro lado; una interfaz hombre - máquina o HMI ("human machine interface") es el aparato que presenta los datos a un operador generalmente humano y a través del cual éste controla y supervisa el proceso.

La industria de HMI nació esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLCs y otros mecanismos de control. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuyen a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual, los sistemas Scada lo hacen de manera automática.

El PLC no tiene una manera estándar de presentar la información al operador. La obtención de los datos por el sistema Scada parte desde el PLC o desde otros controladores y se realiza por medio de algún tipo de red, posteriormente esta información es combinada y formateada.

Un HMI puede tener también vínculos con una base de datos para proporcionar las tendencias, los datos de diagnóstico y manejo de la información así como un cronograma de procedimientos de mantenimiento, información logística, esquemas detallados para un sensor o máquina en particular, incluso sistemas expertos con guía de resolución de problemas.²⁸

Desde cerca de 1998, virtualmente todos los productores principales de PLC ofrecen integración con sistemas HMI/Scada, muchos de ellos usan protocolos de comunicaciones abiertos y no propietarios. Numerosos paquetes de HMI/Scada de terceros ofrecen compatibilidad incorporada con la mayoría de PLCs, incluyendo la entrada al mercado de ingenieros mecánicos, eléctricos y técnicos para configurar estas interfaces por sí mismos, sin la necesidad de un programa hecho a medida escrito por un desarrollador de software.

HMI/Scada es popular debido a esta compatibilidad y seguridad. Ésta se usa desde aplicaciones pequeñas, como controladores de temperatura en un espacio, hasta aplicaciones muy grandes como el control de plantas nucleares.

²⁸

<http://iaci.unq.edu.ar/informacion/..%5Cmaterias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>
Sistemas HMI / Scadas.

2.1. PROGRESO DE LOS SISTEMAS HMI/SCADA.

Los sistemas HMI/Scada utilizan el computador y las tecnologías de comunicación para automatizar el monitoreo y control de procesos industriales. Estos sistemas son partes integrales de la mayoría de los ambientes industriales complejos o muy geográficamente dispersos ya que pueden recoger la información de una gran cantidad de fuentes muy rápidamente, y la presentan a un operador en una forma amigable. Los sistemas HMI/Scada mejoran la eficacia del proceso de monitoreo y control proporcionando la información oportuna para poder tomar decisiones operacionales apropiadas.

Los primeros HMI/Scada eran simplemente sistemas de telemetría que proporcionaban reportes periódicos de las condiciones de campo vigilando las señales que representaban medidas y/o condiciones de estado en ubicaciones de campo remotas. Estos sistemas ofrecían capacidades muy simples de monitoreo y control, sin proveer funciones de aplicación alguna. La visión del operador en el proceso estaba basada en los contadores y las lámparas detrás de paneles llenos de indicadores. Mientras la tecnología se desarrollaba, los ordenadores asumieron el papel de manejar la recolección de datos, disponiendo comandos de control, y una nueva función - presentación de la información sobre una pantalla de CRT²⁹. Los ordenadores agregaron la capacidad de programar el sistema para realizar funciones de control más complejas.

Los primeros sistemas automatizados HMI/Scada fueron altamente modificados con programas de aplicación específicos para atender a requisitos de algún proyecto particular. Como ingenieros de varias industrias asistieron al diseño de estos sistemas, su percepción adquirió las características de su propia industria. Proveedores de sistemas de software HMI/Scada, deseando reutilizar su trabajo previo sobre los nuevos proyectos, perpetuaron esta imagen de industria-específicos por su propia visión de los ambientes de control con los cuales tenían experiencia. Solamente cuando nuevos proyectos requirieron funciones y aplicaciones adicionales, hizo que los desarrolladores de sistemas HMI/Scada tuvieran la oportunidad de desarrollar experiencia en otras industrias.

²⁹ CRT: “Cathode Ray Tube” en español: Tubo de Rayos Catódicos. (Pantallas).

Hoy, los proveedores de HMI/Scada están diseñando sistemas que son pensados para resolver las necesidades de muchas industrias, con módulos de software específicos disponibles para proporcionar las capacidades requeridas comúnmente. No es inusual encontrar software HMI/Scada comercialmente disponible adaptado para procesamiento de papel y celulosa, industrias de aceite y gas, hidroeléctricas, gerenciamiento y provisión de agua, control de fluidos, etc.

Puesto que los proveedores de HMI/Scada aún tienen tendencia en favor de algunas industrias sobre otras, los compradores de estos sistemas a menudo dependen del proveedor para una comprensiva solución a su requisito, y generalmente procurar seleccionar un vendedor que pueda ofrecer una completa solución con un producto estándar que esté apuntado hacia las necesidades específicas del usuario final. Si selecciona a un vendedor con experiencia limitada en la industria del comprador debe estar preparado para asistir al esfuerzo de ingeniería necesario para desarrollar el conocimiento adicional de la industria requerido por el vendedor para poner con éxito el sistema en ejecución.

La mayoría de los sistemas HMI/Scada que son instalados hoy se está convirtiendo en una parte integral de la estructura de gerenciamiento de la información corporativa. Estos sistemas ya no son vistos por la gerencia simplemente como herramientas operacionales, sino como un recurso importante de información. En este papel continúan sirviendo como centro de responsabilidad operacional, pero también proporcionan datos a los sistemas y usuarios fuera del ambiente del centro de control que dependen de la información oportuna en la cual basan sus decisiones económicas cotidianas. La mayoría de los vendedores principales de HMI/Scada han reconocido esta tendencia, y están desarrollando rápidamente métodos eficientes para hacer disponibles los datos, mientras protegen la seguridad y funcionamiento del sistema. La arquitectura de los sistemas de hoy integra a menudo muchos ambientes de control diferentes, tales como tuberías de gas y aceite, en un solo centro de control.

Para alcanzar un nivel aceptable de tolerancia de fallas con estos sistemas, es común tener ordenadores redundantes operando en paralelo en el centro primario de control y un sistema de reserva del mismo situado en un área geográficamente distante.

Esta arquitectura proporciona la transferencia automática de la responsabilidad del control de cualquier ordenador que pueda llegar a ser inasequible por cualquier razón, a una computadora de reserva en línea, sin interrupción significativa de las operaciones.³⁰

2.2. COMPONENTES DE UN HMI / SCADA.

Los tres componentes de un sistema HMI / SCADA son:

- a) Múltiples unidades de terminal remota (RTU o Estaciones Externas).
- b) Estación Maestra y Computador con HMI.
- c) Infraestructura de Comunicación.

2.2.1. UNIDAD TERMINAL REMOTA.

La RTU se conecta al equipo físicamente y lee los datos de estado como los estados abierto/cerrado desde una válvula o un intercambiador, lee las medidas como presión, flujo, voltaje o corriente. Por el equipo el RTU puede enviar señales que pueden controlarlo: abrirlo, cerrarlo, intercambiar la válvula o configurar la velocidad de la bomba.

La RTU puede leer el estado de los datos digitales o medidas de datos analógicos y envía comandos digitales de salida o puntos de ajuste analógicos.

Una de las partes más importantes de la implementación de HMI/Scadas son las alarmas.

Una alarma es un punto de estado digital que tiene cada valor normal o Alarma. La alarma se puede crear en cada paso que los requerimientos lo necesiten. Un ejemplo de una alarma es la luz de "tanque de combustible vacío" del automóvil. El operador de HMI/Scada pone atención a la parte del sistema que lo requiera por la alarma. Pueden enviarse por correo electrónico o mensajes de texto con la activación de una alarma, alertando al administrador o incluso al operador de HMI/Scada.

La Unidad Terminal Remota puede constar de un autómata programable o PLC.

30

http://www.infopl.net/Documentacion/Docu_SCADA/infopl.net_Introduccion_Sistemas_SCADA.pdf
Desarrollo histórico de los sistemas Scada.

2.2.2. ESTACIÓN MAESTRA.

El término "Estación Maestra" se refiere a los servidores y el software responsable para comunicarse con el equipo de campo como los PLCs, en estos se encuentra el software HMI corriendo para las estaciones de trabajo en el cuarto de control, o en cualquier otro lado.

En un sistema HMI/Scada pequeño, la estación maestra puede estar en un solo computador; en un sistema a gran escala la estación maestra puede incluir muchos servidores, aplicaciones de software distribuido y sitios de recuperación de desastres.

Un HMI/Scada usualmente presenta la información al personal operativo de manera gráfica, en la forma de un diagrama de representación. Esto significa que el operador puede ver un esquema que representa la planta que está siendo controlada. Por ejemplo un dibujo de una bomba conectada a la tubería puede mostrar al operador cuanto fluido esta siendo bombeado desde la bomba a través de la tubería en un momento dado. El operador puede cambiar el estado de la bomba a apagado. El software HMI mostrará el promedio de fluido en la tubería decrementándose en tiempo real. Los diagramas de representación pueden consistir en gráficos de líneas y símbolos esquemáticos para representar los elementos del proceso, o pueden consistir en fotografías digitales de los equipos sobre los cuales se animan las secuencias.

El paquete HMI para el sistema Scada típicamente incluye un programa de dibujo con el cual los operadores o el personal de mantenimiento del sistema usan para cambiar la manera que estos puntos son representados en la interfaz. Esta representación puede ser tan simple como luces de tráfico en pantalla, los cuales representan el estado actual de un campo en el tráfico actual, o tan complejo como una pantalla de multiproyector representando posiciones de todos los elevadores en un rascacielos o todos los trenes de una vía férrea.

Un ejemplo gráfico de un sistema temporizado HMI de una planta de embotellado es, como el que se presenta a continuación:

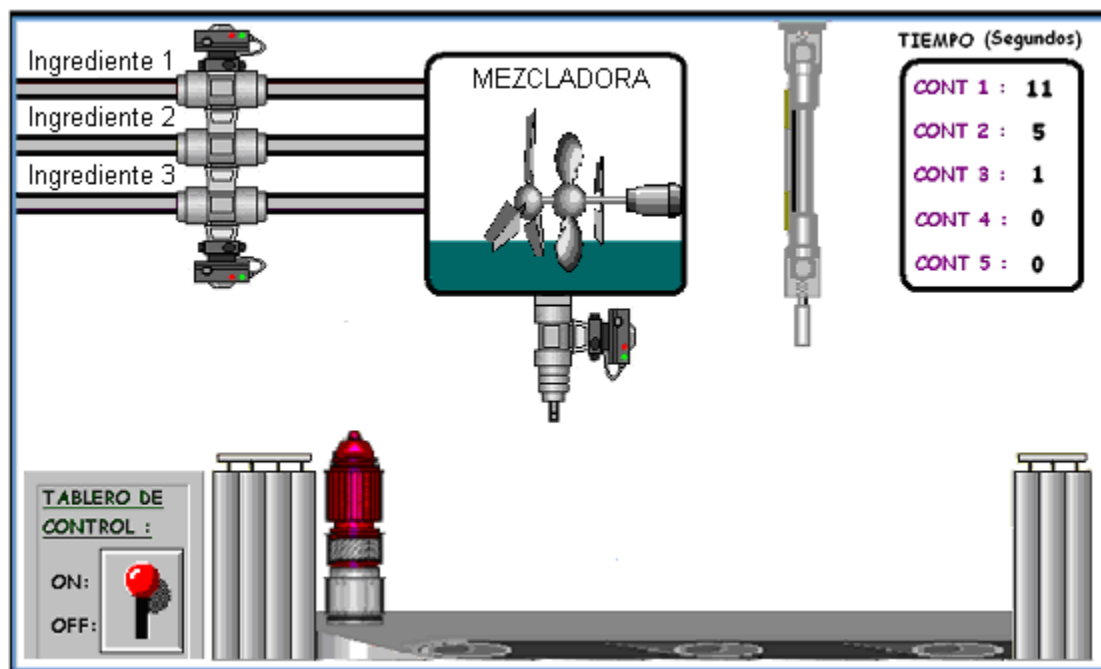


Figura 2.2 Ejemplo de un HMI implementado en Lookout (NI³¹).

Inicialmente más plataformas abiertas como Linux que no eran ampliamente usados, se usan debido al altamente dinámico ambiente de desarrollo y porque un cliente que tiene la capacidad de acomodarse en el campo del hardware y mecanismos a ser controlados que usualmente se venden UNIX o con licencias OpenVMS. Hoy todos los mayores sistemas son usados en los servidores de la estación maestra así como en las estaciones de trabajo HMI.

2.2.3. INFRAESTRUCTURA DE COMUNICACIÓN.

Los sistemas HMI/Scada tienen tradicionalmente una combinación de frecuencias y señales directas seriales o conexiones de modem para conocer los requerimientos de comunicaciones, incluso Ethernet e IP son también frecuentemente usadas en sitios muy grandes como ferrocarriles y estaciones de poder.

Los RTUs pueden ser requeridos para operar ellos mismos su propio control por asuntos relativos a la seguridad. El software de la estación maestra requiere hacer mas análisis de datos antes de ser presentados a los operadores, incluyendo análisis históricos y análisis asociados con los requerimientos de la

³¹ NI "National Instruments" Corporación que desarrolla software para sistemas HMI/Scada.

industria particular. Los requerimientos de seguridad están ahora siendo aplicados en los sistemas como un todo e incluso el software de la estación maestra debe conocer los estándares más fuertes de seguridad para algunos mercados.

En ciertas instalaciones los costos que pueden resultar las fallas de un sistema de control es extremadamente alto. Posiblemente alguna vida podría ser perdida. El Hardware del sistema Scada es generalmente áspero para resistir temperatura, vibración y voltajes extremos pero en estas instalaciones la fiabilidad es aumentada teniendo hardware redundante y varios canales de comunicación.

Una parte que falla puede ser fácilmente identificada y su funcionalidad puede ser automáticamente tomada por un hardware de backup. La parte que falle puede ser reemplazada sin interrumpir el proceso. La confianza en cada sistema puede ser calculado estadísticamente y este estado es el significado de tiempo de falla, el cual es una variable que acumula tiempos entre fallas.

El resultado calculado significa que el tiempo de fallas de sistemas de alta fiabilidad puede ser de siglos.

2.3. PRESTACIONES DE UN HMI / SCADA.

Un paquete HMI/Scada debe estar en disposición de ofrecer las siguientes prestaciones:³²

- a) Posibilidad de crear paneles de alarma, que exigen la presencia del operador para reconocer una parada o situación de alarma, con registro de incidencias.
- b) Generación de históricos de señal de planta, que pueden ser volcados para su proceso sobre una hoja de cálculo.
- c) Ejecución de programas, que modifican la ley de control, o incluso anular o modificar las tareas asociadas al autómata, bajo ciertas condiciones.
- d) Posibilidad de programación numérica, que permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

³² <http://www.automatas.org/redes/scadas.htm> Prestaciones, Requisitos y Módulos de los Sistemas HMI / Scadas.

Con ellas se pueden desarrollar aplicaciones para ordenadores con captura de datos, análisis de señales, presentaciones en pantalla, envío de resultados a disco e impresora, etc.

Además todas estas acciones se llevan a cabo mediante un paquete de funciones que incluye zonas de programación en un lenguaje de uso general, lo cual confiere una potencia muy elevada y una gran versatilidad. Algunos HMI/Scadas ofrecen librerías de funciones para lenguajes de uso general que permiten personalizar de manera muy amplia la aplicación que desee realizarse.

2.4. REQUISITOS DE UN HMI / SCADA.

Un sistema HMI/Scada debe cumplir varios objetivos para que su instalación sea perfectamente aprovechada:

- a) Deben ser sistemas de arquitectura abierta, capaces de crecer o adaptarse según las necesidades cambiantes de la empresa.
- b) Deben comunicarse con total facilidad y de forma transparente al usuario con el equipo de planta y con el resto de la empresa (redes locales y de gestión).
- c) Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware y fáciles de utilizar con interfaces amigables con el usuario.

2.5. MÓDULOS DE UN HMI / SCADA.

Los módulos o bloques software que permiten las actividades de adquisición, supervisión y control de datos son los siguientes:

- a) Configuración: permite al usuario definir el entorno de trabajo de su HMI/Scada, adaptándolo a la aplicación particular que se desea desarrollar.
- b) Interfaz gráfico de operador: proporciona al operador las funciones de control y supervisión de la planta. El proceso se representa mediante sinópticos gráficos almacenados en el ordenador de proceso y generados desde el editor incorporado en el HMI/Scada o importados desde otra aplicación durante la configuración del paquete.
- c) Módulo de proceso: ejecuta las acciones de mando preprogramadas a partir de los valores actuales de variables leídas.

- d) Gestión y archivo de datos: se encarga del almacenamiento y procesamiento ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.
- e) Comunicaciones: se encarga de la transferencia de información entre la planta y la arquitectura hardware que soporta el HMI/Scada y entre ésta y el resto de elementos informáticos de gestión.

2.6. CONCEPTOS ASOCIADOS A UN HMI / SCADA.

2.6.1. TIEMPO REAL.

Se refiere a la capacidad del ordenador en programas de procesamiento de datos para que siempre esté listo para procesar y proporcionar los resultados dentro de un tiempo especificado. Significa que un sistema reacciona a los eventos externos dentro de un tiempo especificado en un 100% de los casos. Si los tiempos concretos de reacción pueden superarse en ciertos casos, como en sistemas no críticos, hablamos de "tiempo real suave".

Modalidad de funcionamiento de un sistema de proceso de datos que controla una actividad en curso, con un tiempo de respuesta prácticamente nulo a la recepción de las señales de entrada. Cuando una acción realizada en el ordenador progresa paralelamente al tiempo del "mundo real", se dice que la acción ocurre en tiempo real.

2.6.2. HARDWARE EN SISTEMAS DE SUPERVISIÓN.

El hecho es que las tareas automatizadas de control, visualización y computación pueden ser efectuadas por PLCs (conectados en red mediante los módulos adecuados) mejor que con sistemas exclusivos de control basados en PC. Lo que finalmente es práctico, depende de un gran número de factores y la mayoría deben ser considerados individualmente para cada proyecto de automatización.

Los actuales conocimientos y preferencias del usuario pueden jugar un mayor papel que la pura potencia del ordenador. Los factores cruciales son los atributos de capacidad en tiempo real y las propiedades de seguridad que hasta ahora han sido fuertemente asociadas con el PLC, aunque el PC también puede disponer de la característica de capacidad en tiempo real. Un sistema de control es

inconcebible sin capacidad en tiempo real. Es común en sistemas de control por ordenador tener que elegir, según las características del sistema a supervisar, entre el PLC o el PC. Se debe elegir aquel hardware que mejor se adapte a las necesidades del sistema a supervisar.

Los controladores lógicos programables, en la mayoría de los casos, están diseñados específicamente para ser empleados en ambientes industriales exigentes y han sido continuamente desarrollados de forma que sus sistemas operativos en tiempo real representan su mayor virtud. Ellos son y seguirán siendo, no obstante, la primera elección para todo control de tareas críticas o extremas por su rendimiento, en los que un PC podría estar simplemente "sobrecargado" debido al trabajo que le pueden suponer otras tareas de ámbito común, como la gestión y visualización de datos, accesos a periféricos, bases de datos, etc.

Si además del control de tareas, se necesita un procesamiento de datos, trabajo en red o visualización (una aplicación HMI/Scada), un sistema basado en PC debe ser tomado en consideración.³³

2.6.3. SOFTWARE DE LOS SISTEMAS HMI / SCADA.

Los primeros sistemas existentes tendieron a ser propietarios y muy especializados, y donde fueron utilizados sistemas operativos de fines generales, tendieron a ser modificados ampliamente. Esto debido a que los requisitos de HMI/Scada superaban los límites de la tecnología disponible en el momento y por razones de desempeño ya que tendieron a proporcionar sistemas gráficos por encargo, a usar bases de datos en tiempo real (con gran parte de la base de datos en memoria) y a menudo el hardware debió ser modificado para estos requisitos particulares.

Posteriormente, Unix comenzó a ser el sistema operativo de más frecuente elección. En épocas recientes Windows NT ha alcanzado alta aceptación dentro de la comunidad Scada, aunque los sistemas muy grandes siguen siendo en la

33

http://www.infopl.net/Documentacion/Docu_SCADA/infopl.net_Introduccion_Sistemas_SCADA.pdf

Hardware y software de los sistemas HMI/Scada.

mayor parte de los casos estaciones de trabajo Unix, las cuales son más veloces en sus respuestas.

Actualmente la industria se está desarrollando claramente hacia estándares abiertos: ODBC, INTEL PC, sistemas estándares de gráficos, e ínter conectividad a sistemas de computación corrientes.

En años recientes ha aparecido en el mercado un importante número de sistemas Scada sobre plataformas INTEL PC, ya que éstas están aumentando rápidamente su capacidad y desempeño.

Ejemplos de ellos son Citect, FIX de Intellution, KEPware y Wonderware.

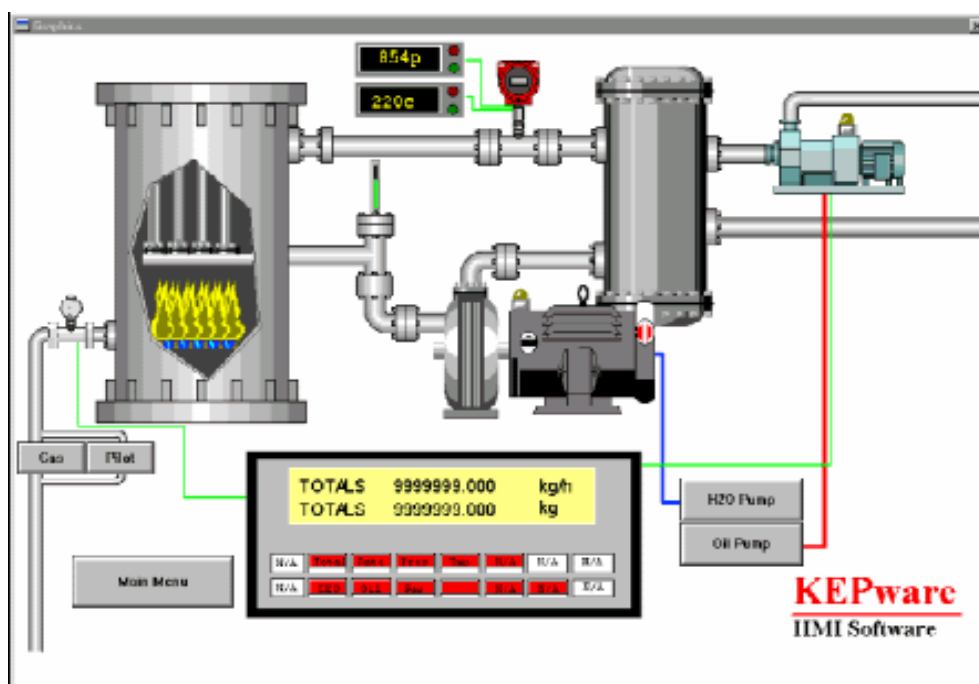


Figura 2.3 Presentación de una planta implementada en KEPware.³⁴

2.6.4. TARJETAS DE EXPANSIÓN.

Como el sistema operativo sólo puede proporcionar respuestas suaves en tiempo real lo más simple es emplear extensiones hardware para las tareas críticas (placas de expansión PC) y soluciones software para el resto de tareas. Esto nos lleva a una compatibilidad con futuros sistemas operativos y una solución

34

http://www.infopl.net/Documentacion/Docu_SCADA/infopl.net_Introduccion_Sistemas_SCADA.pdf
Hardware y software de los sistemas HMI/Scada.

totalmente factible actualmente. Estas tarjetas de expansión asumen las tareas críticas en tiempo real que el ordenador no puede atender, se está hablando de tarjetas que incorporan DSPs (Procesadores de Señales Digitales) o microcontroladores y que aportan una ayuda a la anterior “sobrecarga” mencionada para los ordenadores.³⁵

2.6.5. ESTRUCTURA ABIERTA.

Aún no se ha establecido un estándar para poseer extensiones compatibles en tiempo real de sistemas operativos. De una forma estrictamente determinante, los sistemas estándar actuales deben ser modificados de forma general, así que la principal ventaja de un sistema basado en PC - su estructura abierta – puede llegar a ser un inconveniente.

La estructura abierta permite a la empresa o al desarrollador más libertad en la elección de la herramienta adecuada para el análisis, diseño y programación del sistema HMI/Scada. La solución comienza a ser propietaria nuevamente (cada empresa ofrece su solución) y la conversión a futuras generaciones de sistemas operativos lo hace más difícil.

2.6.6. COMPUTADORES INDUSTRIALES.

Los computadores industriales deben tener funciones de vigilancia dedicadas al control y supervisión, deben trabajar en ambientes pesados con equipos electrónicos especiales manteniendo la seguridad con respecto al ruido o interferencias, deben ser de diseño y tecnología abierta, deben brindar mayor flexibilidad y brindar diferentes configuraciones.

Además deben ser confiables y tolerantes a fallos comunes sin la necesidad de detener el proceso que están ejecutando. A nivel de hardware existen computadoras dobles, con doble procesador, memorias duplicadas y dos fuentes de poder. A nivel de software el sistema operativo tiene secciones de vigilancia supervisando permanentemente el proceso, cuando detecta anomalías informa directamente al operario en forma de alarmas.

También deben brindar seguridad, este parámetro depende estrictamente del ambiente en el cual se desempeñará el tipo de producto que se controla y de la

³⁵ <http://personal.redestb.es/efiguera/memoria.htm> Tarjetas de Expansión y Estructura Abierta.

tarea específica que cumple. Las computadoras industriales deben responder a estándares y normas de seguridad de dispositivos eléctricos y electrónicos.

Las PC industriales dependen del hardware que contienen internamente y de la estabilidad del sistema operativo.



Figura 2.4 PC Industrial.

Existen varios tipos de PC industriales:

- a) Compactos.- Son rígidas, tienen un desempeño excelente, son de poca flexibilidad para realizar tareas básicas.
- b) De montaje en panel.- Son flexibles y permiten el uso de Touch screen³⁶, tienen fácil interfaz y son las más utilizadas en la industria.
- c) De montaje en Rack.- Permiten capacidad variable, carcasas removibles, aceptan varios procesadores, tienen sistemas de protección y accesos a slots.
- d) Portátiles.- Realizan tareas de control, no son como las conocidas portátiles o Laptop porque su peso y tamaño son más grandes, pero son transportables y maniobrables.

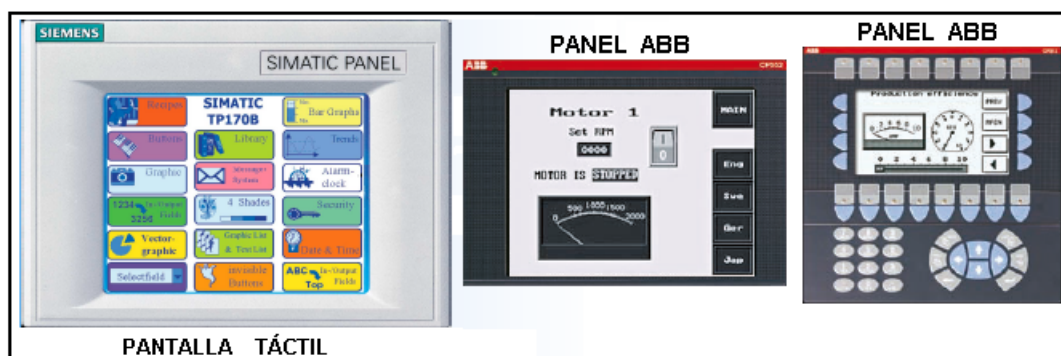


Figura 2.5 Tipos de paneles industriales.

³⁶ Touch screen en español: pantalla táctil.

En cuanto al software, trabajan con Windows 95, 98, 2000 y XP, tienen gran cantidad de drivers, su arquitectura abierta hace que se incremente su compatibilidad. Las PC industriales son el intermediario entre los sistemas informáticos y las soluciones concretas que ofrecen los dispositivos programables como PLCs, Fieldpoints, micro PLCs o controladores especializados.

2.7. AUTÓMATA LÓGICO PROGRAMABLE.

Un PLC es un “Programmable Logic Controller” que en español significa: Controlador Lógico Programable. Un PLC es un dispositivo usado para controlar, este control se realiza sobre la base de una lógica definida a través de un programa. Es un equipo electrónico programable en lenguaje no informático, diseñado para controlar en tiempo real y en ambiente de tipo industrial, procesos secuenciales. Un PLC trabaja en base a la información recibida por los captadores y el programa lógico interno, actuando sobre los accionadores de la instalación.³⁷



Figura 2.7 PLCs de la familia MicroLogix 1500.

Los autómatas programables son máquinas secuenciales que ejecutan correlativamente las instrucciones indicadas en el programa de usuario almacenado en su memoria, generando unas ordenes o señales de mando a partir de las señales de entrada leídas de la planta (aplicación): al detectarse cambios en las señales, el autómata reacciona según el programa hasta obtener

³⁷

<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMHI/PAGINA%20PRINCIPAL/PLC/plc.htm>
Introducción y definición de un PLC.

las ordenes de salida necesarias. Esta secuencia se ejecuta continuamente para conseguir el control actualizado del proceso.

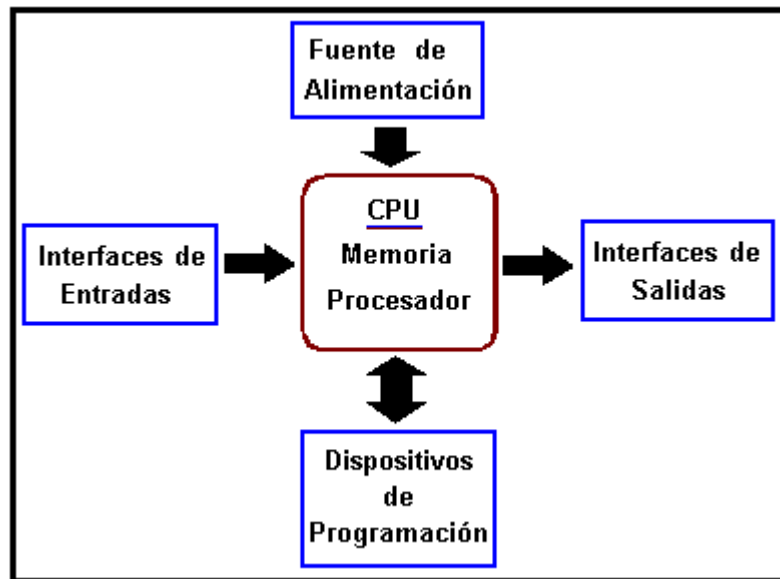


Figura 2.8 Estructura de un Controlador Lógico Programable.³⁸

La secuencia básica de operación del autómatas se puede dividir en tres fases principales:

- a) Lectura de señales desde la interfaz de entradas.
- b) Procesado del programa para obtención de las señales de control.
- c) Escritura de señales en la interfaz de salidas.

2.7.1. FUNCIONAMIENTO.

A fin de optimizar el tiempo, la lectura y escritura de las señales se realiza a la vez para todas las entradas y salidas. Entonces las entradas leídas de los módulos de entrada se guardan en una memoria temporal (Imagen entradas). A esta acude la CPU en la ejecución del programa, y según se va obteniendo las salidas, se guardan en otra memoria temporal (imagen de salida). Una vez ejecutado el programa completo, estas imágenes de salida se transfieren todas a la vez al módulo de salida.

El autómatas realiza también otra serie de acciones que se van repitiendo periódicamente, definiendo un ciclo de operación.

³⁸ <http://www.profesores.frc.utn.edu.ar/industrial/sistemasinteligentes/index.html>
Controlador Lógico Programable PLC.

El usuario ingresa el programa a través del dispositivo adecuado (un cargador de programa o PC) y éste es almacenado en la memoria de la CPU.

La CPU, que es el "cerebro" del PLC, procesa la información que recibe del exterior a través de la interfaz de entrada y de acuerdo con el programa, activa una salida a través de la correspondiente interfaz de salida.

Evidentemente, las interfaces de entrada y salida se encargan de adaptar las señales internas a niveles de la CPU. Por ejemplo, cuando la CPU ordena la activación de una salida, la interfaz adapta la señal y acciona un componente (transistor, relé, etc).³⁹

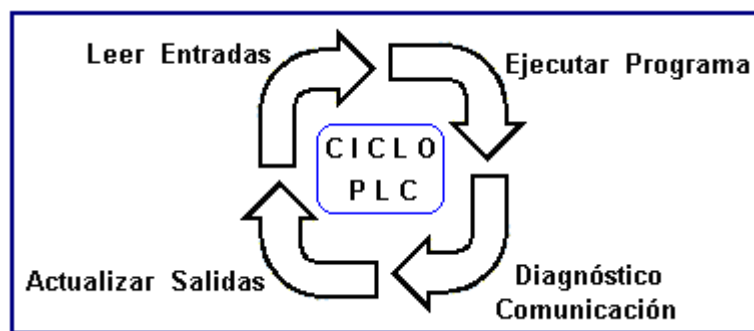


Figura 2.9 Ciclo del PLC.

El ciclo de funcionamiento se divide en dos partes llamados *Proceso Inicial* y *Ciclo de Operación*.

2.7.1.1. Proceso Inicial.

Antes de entrar en el ciclo de operación el autómata realiza una serie de acciones comunes, que tratan fundamentalmente de inicializar los estados del mismo y chequear el hardware. Estas rutinas de chequeo, incluidas en el programa monitor ROM, comprueban:

- a) El bus de conexiones de las unidades de E/S.
- b) El nivel de la batería, si esta existe.
- c) La conexión de las memorias internas del sistema.
- d) El módulo de memoria exterior conectado, si existe.

³⁹

Si se encontrara algún error en el chequeo, se activaría el LED de error y quedaría registrado el código del error.

Comprobadas las conexiones, se inicializan las variables internas:

- a) Se ponen en OFF las posiciones de memoria interna (excepto las mantenidas o protegidas contra perdidas de tensión)
- b) Se borran todas las posiciones de memoria imagen E/S.
- c) Se borran todos los contadores y temporizadores (excepto los mantenidos o protegidos contra perdidas de tensión).

Transcurrido el *Proceso Inicial* y si no han aparecido errores el autómata entra en el *Ciclo de Operación*.

2.7.1.2. Ciclo de operación.

Este ciclo puede considerarse dividido en tres bloques que son: proceso común, ejecución del programa, servicio a periféricos.

Proceso común:

En este primer bloque se realizan los chequeos cíclicos de conexiones y de memoria de programa, protegiendo el sistema contra:

- a) Errores de hardware (conexiones E/S, ausencia de memoria de programa).
- b) Errores de sintaxis (programa imposible de ejecutar).
- c) Niveles de tensión de alimentación.
- d) Estado de la batería si existe.
- e) Buses de conexión con las interfaces.
- f) El chequeo de la memoria de programa comprueba la integridad de la misma y los posibles errores de sintaxis y gramática:
- g) Mantenimiento de los datos, comprobados en el "checksum".
- h) Existencia de la instrucción END de fin de programa.
- i) Estructura de saltos y anidamiento de bloque correctas.
- j) Códigos de instrucciones correctas.

Ejecución del programa:

En este segundo bloque se consultan los estados de las entradas y de las salidas y se elaboran las órdenes de mando o de salida a partir de ellos.

El tiempo de ejecución de este bloque de operaciones es la suma del tiempo de acceso a interfaces de E/S más el tiempo de escrutación de programa.

Y a su vez esto depende del número, ubicación de las interfaces de E/S y de la longitud del programa y tipo de CPU que lo procesa.

Servicio a periféricos:

Este tercer y último bloque es únicamente atendido si hay pendiente algún intercambio con el exterior, en caso de haberlo la CPU le dedica un tiempo limitado de 1 a 2ms, en atender el intercambio de datos, si este tiempo no fuera suficiente el servicio queda interrumpido hasta el siguiente ciclo.

2.7.2. FUNCIONES DEL PLC.

Dentro de las funciones del PLC se puede mencionar:

- Adquirir datos del proceso por medio de las entradas digitales y analógicas.
- Tomar decisiones en base a reglas programadas.
- Almacenar datos en memoria.
- Generar ciclos de tiempo.
- Realizar cálculos matemáticos.
- Actuar sobre dispositivos externos mediante las salidas digitales y analógicas.
- Comunicarse con otros sistemas externos.

2.7.3. UNIDADES FUNCIONALES.

Un controlador lógico programable se compone de cuatro unidades funcionales: unidad de entradas, unidad de salidas, unidad lógica, unidad de memoria.⁴⁰

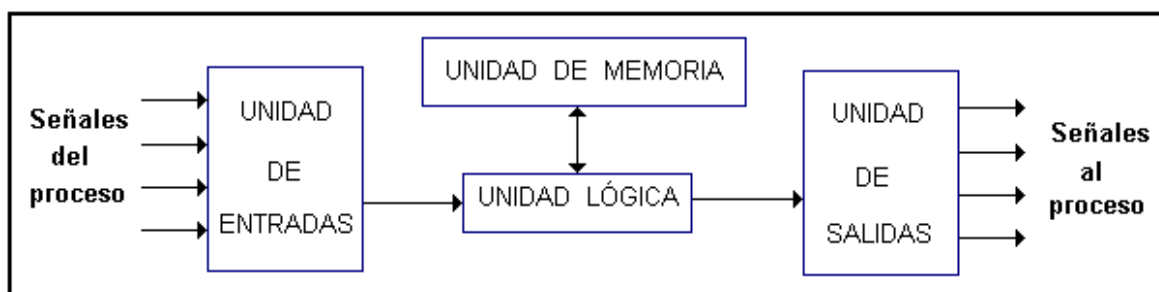


Figura 2.10 Unidades Funcionales.

40

2.7.3.1. Unidad de Entradas.

Proporciona el aislamiento eléctrico necesario y realiza el acondicionamiento de las señales eléctricas de voltaje, provenientes de los switches de contactos ON-OFF de terreno. Las señales se adecuan a los niveles lógicos de voltaje de la unidad lógica.

2.7.3.2. Unidad de Salidas.

Acepta las señales lógicas provenientes de la unidad lógica, en los rangos de voltaje que le son propios y proporciona la aislamiento eléctrico a los switches de contactos que se comandan hacia terreno. Las unidades de entrada/salida del PLC, son funcionalmente iguales a los bancos de relés, que se empleaban en los antiguos controladores lógicos de tipo tambor, la diferencia radica en que las unidades de entrada/salida de los PLC son de estado sólido. La eliminación de contactos mecánicos se traduce en una mayor velocidad de operación y mayor tiempo entre fallas.

2.7.3.3. Unidad Lógica.

Es el corazón de un PLC, está basada en un microprocesador que ejecuta las instrucciones programadas en memoria para desarrollar los esquemas de control lógico que se especifican.

Algunos equipos antiguos implementan la unidad lógica en base a elementos discretos: compuertas, contadores, etc.

2.7.3.4. Unidad de memoria.

Almacena el código de mensajes o instrucciones que ejecuta la unidad lógica. La memoria se divide en PROM o ROM y RAM.

- a) ROM: Memoria de sólo lectura (Read Only Memory). Memoria no volátil que puede ser leída pero no escrita. Es utilizada para almacenar programas y datos necesarios para la operación de un sistema basado en microprocesadores.
- b) RAM: Memoria de acceso aleatorio (Random Access Memory). Memoria volátil que puede ser leída y escrita según sea la aplicación. Cualquier posición de memoria puede ser accesada en cualquier momento.

Por medio de ellas, se puede utilizar un PLC en procesos diferentes sin necesidad de readecuar o transformar el equipo, sólo se debe modificar el programa. Para el control de un proceso, se pueden almacenar varias recetas en la memoria y acceder aquella que interesa.

Las PROM o ROM almacenan los programas permanentes que coordinan y administran los recursos del equipo.

La RAM guarda los programas de aplicación que pueden sufrir modificaciones. Esta memoria es respaldada con baterías, con el propósito de no perder la información al existir cortes de fluido eléctrico.

El sistema opera a través de la interacción con el procesador y la memoria. Cuando se enciende el equipo, el procesador lee la primera palabra de código almacenada en memoria y la ejecuta. Una vez que termina de ejecutar la instrucción leída, busca en memoria la siguiente instrucción y así sucesivamente hasta que se completa la tarea. Esta operación se llama ciclo de búsqueda-ejecución.

2.7.4. MÓDULOS DE UN PLC.

Los módulos básicos de un PLC son: fuente de poder, CPU, interfaces de entrada y salida. Otros módulos importantes son:

2.7.4.1. Módulo de comunicaciones.

Permite la conexión del PLC a otros sistemas de información, tales como computadores y otros PLC, existen por ejemplo redes tipo DataHiway para establecer una red de PLC conectados a un computador Host, utilizada comúnmente en sistemas de control distribuido.

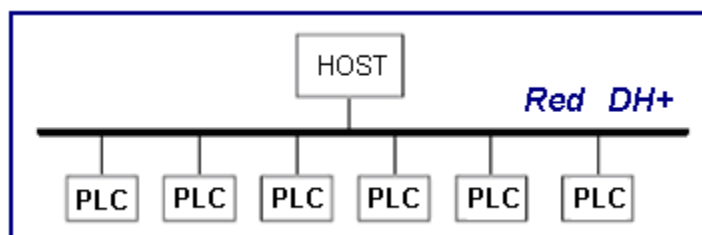


Figura 2.11 Red tipo DataHiway.

2.7.4.2. Módulo de control de redundancia.

Son utilizados para asegurar la operación de un módulo redundante en caso de fallas. Generalmente se utiliza redundancia para el módulo de fuente de alimentación y el CPU.

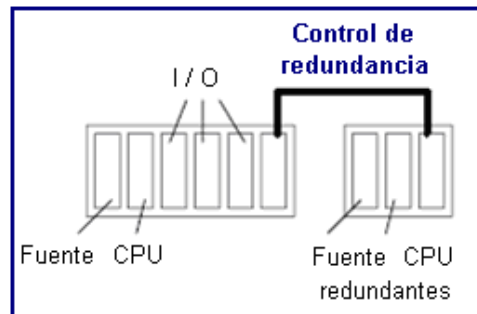


Figura 2.12 Módulo de control de redundancia.

2.7.4.3. Módulo para conexión de rack remoto.

En muchas aplicaciones los sensores y los actuadores están localizados a gran distancia del PLC. En estos casos se utilizan los racks remotos, los que son conectados por medio de un cable al rack central del PLC. Se consiguen distancias de 300 metros.

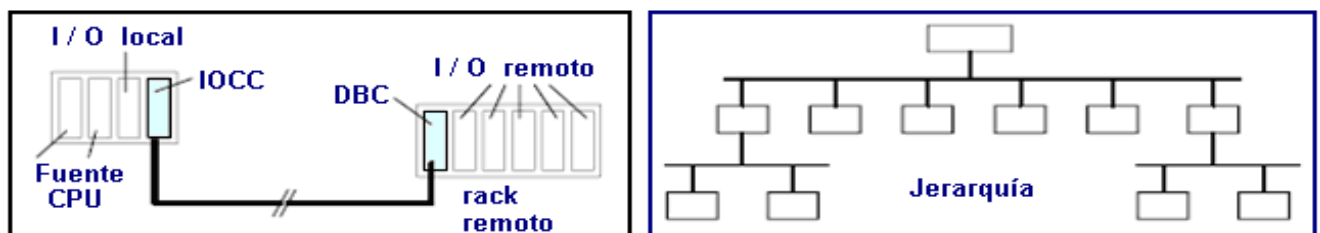


Figura 2.13 Módulo para conexión de rack remoto y jerarquía.

Para establecer esta comunicación se utiliza un módulo denominado canal controlador de entradas y salidas (IOCC) en el rack local y otro llamado controlador de base (DBC) en el rack remoto, al que se le puede conectar otro rack remoto, estableciéndose así una arquitectura distribuida con distintos niveles de jerarquía.

2.7.4.4. Módulo de interfaz hombre-máquina.

Se utilizan para establecer la comunicación entre el PLC y el usuario. En la mayoría de los casos se emplea un computador PC conectado serialmente, desde

el cual se puede programar el PLC y ver los estados de los registros internos y los puntos de entrada/salida, en otros casos se usa un hand held monitor, que es un dispositivo pequeño con teclas funcionales y pantalla de caracteres.

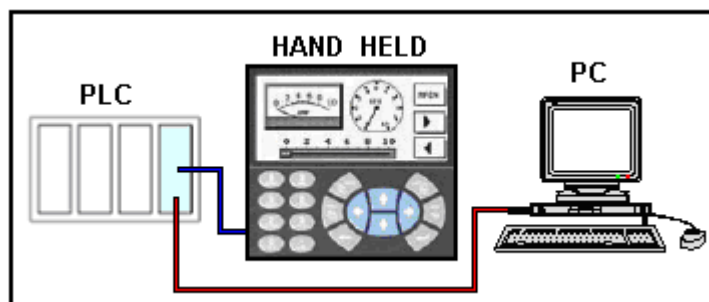


Figura 2.14 Módulo HMI (monitor Hand Held).

2.7.4.5. Módulos de Entrada y Salida.

La información entra y sale del CPU a través de estos módulos. Las terminales de los módulos de entrada reciben las señales que el CPU debe procesar. Las terminales de los módulos de salida proveen voltajes y/o corrientes para energizar los actuadores y dispositivos señalizadores.

Las señales de entrada pueden provenir de dispositivos como: interruptores limitadores, sensores fotoeléctricos, sensores de proximidad, interruptores de presión, interruptores de nivel, interruptores de temperatura, etc.

Las señales de salida pueden estar dirigidas a diferentes tipos de actuadores como: válvulas, arrancadores de motores, solenoides, hornos, alarmas, señalizadores, relés de control, contadores, etc.

Existen varios tipos de módulos de entradas/salidas, que dependen del tipo de señal que reciben o producen como:

- a) Módulos de entradas digitales: Convierten las señales digitales externas del proceso al nivel interno del autómeta.
- b) Módulos de salida digitales: Convierten las señales internas del PLC en señales externas adaptadas al proceso.
- c) Módulos de entradas analógicas: Convierte las señales analógicas en señales digitales que autómeta procesa internamente.
- d) Módulos de salida analógicas: Convierten las señales digitales del PLC en señales analógicas para el proceso.

- e) Módulo de simulación: Provee señales de entrada de prueba al PLC para verificar el correcto funcionamiento del programa antes de conectarlo en la planta. Este módulo es útil cuando se tienen pocas señales de entrada; sin embargo, cuando la cantidad de entradas es muy alta y dependen unas de otros, es más difícil simular el programa.⁴¹

2.7.5. PROGRAMADOR / MONITOR (PM).

También se le conoce como dispositivo de interfaz hombre-máquina.

Este se conecta al CPU mediante un cable, después de la programación del CPU, el PM por lo general no es tan necesario para la operación del sistema y puede ser removido, se puede utilizar el mismo PM para diversos dispositivos, más si se toma en cuenta que algunos fabricantes actuales tienen dispositivos especialmente hechos para transportarlos y utilizarlos en el lugar donde se encuentra el PLC o HMI portátiles.

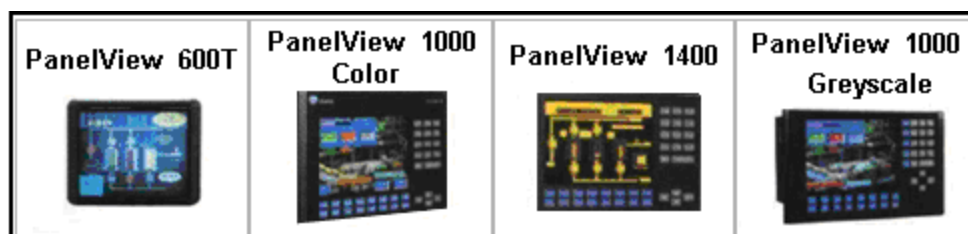


Figura 2.15 Dispositivos de HMI de Allen – Bradley.

2.7.6. VENTAJAS Y DESVENTAJAS DEL PLC.

Algunas de las ventajas de usar un PLC son las siguientes:

- Flexibilidad: Un modelo de PLC puede controlar varias máquinas diferentes cada una con su programa propio, no se requiere un controlador para cada una como sucedía anteriormente. Posibilidad de gobernar varias máquinas con un mismo autómata. Además si por alguna razón la máquina queda fuera de servicio, el autómata sigue siendo útil para otra máquina o sistema de producción.
- Implementación de cambios y corrección de errores: Cuando se decide cambiar un programa o parte de él, se puede realizar desde un dispositivo

⁴¹ <http://www.profesores.frc.utn.edu.ar/industrial/sistemasinteligentes/PLC.html> Módulos de un PLC.

de entrada como un teclado en solo unos minutos; anteriormente se requería realizar de nuevo el alambrado del panel de control. Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir aparatos.

- c) Bajo costo: El incremento de la tecnología ha posibilitado introducir mayor cantidad de funciones en espacios más reducidos y más baratos. Sin embargo esta ventaja podría ser un inconveniente para las pequeñas industrias que deben invertir en un capital inicial para el proyecto de automatización o de HMI/Scada.
- d) Prueba piloto: El PLC programado puede ser probado para evaluarlo en condiciones de laboratorio, de esta forma se realizan las modificaciones necesarias para que trabaje correctamente en el campo.
- e) Velocidad de operación: La velocidad de operación de un programa de PLC es bastante rápida, en general está determinada por el tiempo de escaneo de los datos, que ronda los milisegundos.
- f) Seguridad: Un cambio en el programa del PLC no puede hacerse a menos que éste sea debidamente intervenido; en los sistemas de relés se podían realizar cambios fácilmente sin documentación, que eran olvidados fácilmente.
- g) Menor espacio físico: Ocupación del mínimo espacio físico y además de aumentar la fiabilidad del sistema, al eliminar contactos móviles, los mismos autómatas pueden indicar y detectar averías.
- h) Menor tiempo: Menos tiempo para la puesta en funcionamiento del proceso al quedar reducido el tiempo en la instalación del cableado. Además ganancia de tiempo por la economía de mantenimiento.⁴²

Estas son sólo algunas de las ventajas que presentan los PLC, aunque pueden derivarse otras, de las aplicaciones particulares en un campo específico. También existen desventajas en el uso de estas herramientas:

- a) Tecnología nueva: Puede resultar difícil cambiar el modo de pensar de algunas personas de los relés a un concepto más computarizado como el

⁴²

<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCOMH1/PAGINA%20PRINCIPAL/PLC/plc.htm#Ventajas%20e%20inconvenientes> Ventajas, inconvenientes y campos de aplicación de los PLCs.

PLC. Sin embargo actualmente el uso de las computadoras está bastante extendido en hogares, oficinas e industria y poco a poco la gente se ha ido habituando a su uso.

- b) Aplicaciones de programas fijos: Algunas aplicaciones son de función simple es decir su operación no se cambia nunca o casi nunca. No es una buena inversión un PLC con su capacidad de manejo de múltiples programas y facilidad de variar los mismos, si no se va a utilizar.
- c) Consideraciones ambientales: Las condiciones ambientales en que se desarrollan ciertos procesos, como alta temperatura y vibraciones, producen interferencias en los dispositivos electrónicos del PLC.
- d) Operación segura en fallas: En los sistemas de relés, una falla en la alimentación detiene el sistema y el relé no se restaura automáticamente cuando esta acaba. Esto se puede programar en un PLC, sin embargo algunos programas requieren alimentación para detener el dispositivo. Esta desventaja se puede resolver agregando relés de seguridad al sistema del PLC.
- e) Falta de programadores: Lo que obliga a adiestrar a uno de los técnicos en tal sentido, pero hoy en día ese inconveniente esta solucionado porque las universidades ya se encargan de dicho adiestramiento.

2.7.7. APLICACIONES DE LOS PLC.

El PLC es usado en la actualidad en una amplia gama de aplicaciones de control, muchas de las cuales no eran económicamente posibles hace algunos años. Esto debido al costo efectivo por punto de entrada/salida que ha disminuido con la caída del precio de los microprocesadores y los componentes relacionados y la capacidad de los controladores para resolver tareas complejas de computación y comunicación ha hecho posible el uso de PLC en aplicaciones donde antes era necesario dedicar un computador.

Los PLCs por sus especiales características de diseño tienen un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc. Por tanto su aplicación abarca procesos de fabricación industriales de cualquier tipo, transformaciones industriales, control de instalaciones, etc.

Sus reducidas dimensiones, la extremada facilidad de su montaje, la posibilidad de almacenar los programas para su posterior y rápida utilización, la modificación o alteración de los mismos, hace que su eficacia se aprecie fundamentalmente en procesos en que se producen necesidades tales como:

- a) Control secuencial.
- b) Control de movimiento.
- c) Control, monitoreo y supervisión de procesos.
- d) Administración de datos.
- e) Comunicaciones.
- f) Procesos de producción periódicamente cambiantes y maquinaria de procesos variables.
- g) Instalaciones de procesos complejos y amplios.
- h) Chequeo de programación centralizada de las partes del proceso.

Ejemplos de aplicaciones generales:

Maniobra de máquinas, maquinaria industrial de plástico, máquinas transfer, maquinaria de embalajes, maniobra de instalaciones de: aire acondicionado, calefacción y seguridad; señalización y control en: chequeo de programas y estado de procesos.

2.7.8. REDES INDUSTRIALES.

Existen diferentes medios para que los datos puedan ser intercambiados entre los instrumentos de campo y el ordenador. Muchos de los instrumentos poseen un puerto serie, mediante el cual la información es enviada al ordenador o a otros instrumentos. El uso de GPIB (Buses de Intercambio de Propósito General) permiten transferir datos a través de puertos paralelos, puertos series y redes de instrumentos u ordenadores.

Los controladores lógico programables pueden comunicarse entre sí y otros dispositivos en una gran variedad de redes.

Debido a que el presente trabajo de tesis se realiza con PLC Siemens S7 con un CPU 224, y se programa con STEP 7 MicroWin, se trata de la red tipo Tokeng Ring.

Basándose en la intercomunicación de sistemas abiertos (OSI) de la arquitectura de siete capas, los protocolos se implementan en una red "token ring" (red de anillo con testigo) conforme al estándar PROFIBUS, definido en la Norma Europea EN 50170. Se trata de protocolos asíncronos de caracteres que utilizan un bit de inicio, ocho bits de datos, un bit de paridad par y un bit de parada. Los bloques de comunicación dependen de los caracteres especiales de inicio y de parada, de las direcciones de estación de fuente y de destino, de la longitud de los bloques y de la suma de verificación para garantizar la integridad de los datos. Los protocolos se pueden utilizar simultáneamente en una red sin que interfieran entre sí, con la condición de que usen una misma velocidad de transferencia.

Los protocolos soportan 127 direcciones (0 a 126) en una red. Una red puede comprender 32 maestros máximo. Todos los dispositivos que formen parte de una red deberán tener direcciones unívocas para poder comunicarse entre sí. El ajuste estándar para las unidades de programación Simatic y los PCs con STEP 7-Micro/WIN 32 es la dirección: 0. Los visualizadores de textos TD 200 y los paneles de operador OP3 y OP7 tienen la dirección predeterminada: 1. La dirección estándar de los sistemas de automatización es: 2.

2.7.8.1. Protocolos de Comunicación.

Las CPUs S7-200 soportan uno o varios de los protocolos de comunicación siguientes. Estos protocolos permiten configurar la red conforme al rendimiento y a la funcionalidad que exige la aplicación:

- a) Interfaz punto a punto (PPI).
- b) Interfaz multipunto (MPI).
- c) Process Field Bus (PROFIBUS).
- d) Freeport

Protocolo PPI:

PPI es un protocolo maestro-esclavo, Los maestros envían peticiones a los esclavos y éstos responden.

Los esclavos no inician mensajes, sino que esperan a que un maestro les envíe una petición o solicite una respuesta.

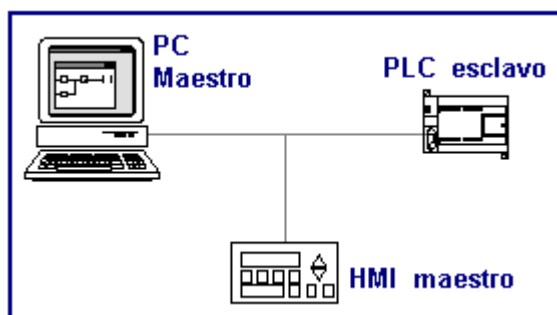


Figura 2.16 Protocolo PPI.

Los maestros se comunican con los esclavos vía un enlace compartido que es gestionado por el protocolo PPI. El protocolo PPI no limita el número de maestros que se pueden comunicar con un mismo esclavo. Sin embargo, la red no puede comprender más de 32 maestros.

Si se selecciona el protocolo PPI Avanzado es posible establecer un enlace lógico entre los aparatos. En este caso cada aparato soporta un número de enlaces limitado. Estando en modo RUN, algunas CPUs S7-200 pueden actuar de estaciones maestras en la red si está habilitado el modo maestro PPI en el programa de usuario. Una vez habilitado este modo las operaciones Leer de la red (NETR) y Escribir en la red (NETW) se podrán utilizar para leer o escribir en otras CPUs S7-200. Mientras actúa de maestro PPI el S7-200 sigue respondiendo en calidad de esclavo a las peticiones de otros maestros.

El protocolo PPI se puede utilizar para la comunicación con todas las CPUs S7-200. Para comunicarse con un módulo de ampliación EM 277 es preciso habilitar el modo PPI Avanzado.

Protocolo MPI:

El protocolo MPI soporta la comunicación maestro-maestro, o bien maestro - esclavo.

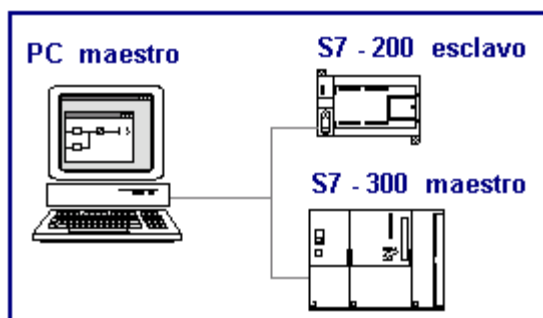


Figura 2.17 Protocolo MPI.

El protocolo MPI no sirve para comunicarse con una CPU S7-200 que actúe de maestra. Los aparatos de la red se comunican a través de enlaces separados (gestionados por el protocolo MPI) entre dos aparatos cualesquiera. La comunicación entre los aparatos se limita a la cantidad de enlaces que soportan las CPU. Los sistemas de automatización S7-300 y S7-400 utilizan las operaciones XGET y XPUT para leer y escribir datos en la CPU S7-200.

Protocolo PROFIBUS:

El protocolo profibus se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas (E/S remotas). Hay numerosos aparatos profibus ofrecidos por diversos fabricantes. Estos aparatos abarcan desde módulos sencillos de entradas o salidas hasta controladores de motores y sistemas de automatización (autómatas programables).

Por lo general, las redes profibus incorporan un maestro y varios esclavos.

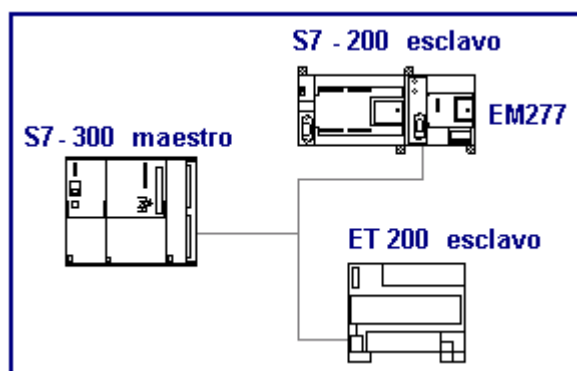


Figura 2.18 Protocolo Profibus.

La configuración del maestro permite detectar los tipos de esclavos conectados, así como sus respectivas direcciones. El maestro inicializa la red y verifica si los esclavos coinciden con la configuración. Continuamente, el maestro escribe los datos de salida en los esclavos y lee de allí los datos de entrada. Una vez que un maestro DP haya configurado correctamente a un esclavo éste último le pertenecerá. Si hay otro maestro en la red, tendrá apenas un acceso muy limitado a los esclavos del primer maestro.

Este protocolo de comunicación es usado para gran capacidad de transmisión de datos.⁴³

La red profibus utiliza el estándar RS-485 con cables de par trenzado. Ello permite interconectar hasta 32 dispositivos en un segmento de la red. Los segmentos pueden tener una longitud máxima de 1200 m, dependiendo de la velocidad de transferencia. Es posible conectar repetidores para poder incorporar más dispositivos en la red o con objeto de utilizar cables más largos. Las redes pueden tener una longitud de hasta 9.600 m, utilizando un total de hasta 9 repetidores (dependiendo de la velocidad de transferencia).

Protoco FREEPORT:

Protocolo definidos por el usuario. La comunicación Freeport es un modo de operación que permite al programa de usuario controlar el puerto de comunicación de la CPU S7-200. Con el modo Freeport se pueden implementar protocolos de comunicación definidos por el usuario para crear enlaces con numerosos dispositivos inteligentes.

El programa de usuario controla el funcionamiento del puerto de comunicación utilizando interrupciones de recepción y de transmisión, así como las operaciones Transmitir mensaje (XMT) y Recibir mensaje (RCV). En modo Freeport, el programa de usuario controla por completo el protocolo de comunicación. El modo Freeport se habilita con la marca SMB30 (puerto 0), estando activo únicamente cuando la CPU está en modo RUN. Cuando la CPU retorna a modo STOP, la comunicación Freeport se detiene y el puerto de comunicación vuelve a utilizar el protocolo PPI normal.

⁴³ Redes, Protocolos e Interfaces de comunicación de los PLCs:

a) <http://www.profesores.frc.utn.edu.ar/industrial/sistemasinteligentes/PLC.html>

b) Manual de usuario de Simatic S7 – 200 y Ayuda de STEP 7 MicroWin 32.

2.7.8.2. Comunicación.

Las comunicaciones industriales son generalmente seriales. El puerto serial es muy extendido y ya sean uno o dos puertos con conectores grandes o pequeños, todas las computadoras lo incorporan actualmente. La comunicación realizada con el puerto serial es una comunicación asincrónica. Para la sincronización de una comunicación se precisa siempre de un bit adicional a través del cual el emisor y el receptor intercambian la señal del pulso. Pero en la transmisión serial a través de un cable de dos líneas esto no es posible ya que ambas están ocupadas por los datos y la tierra.

Para la transmisión serial se necesita: un bit de inicio, un bit de parada y un bit de paridad.

- a) Bit de Inicio: Cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe realizar la lectura de las señales de la línea a distancias concretas de tiempo, en función de la velocidad determinada.
- b) Bit de parada: Indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.
- c) Bit de Paridad: Con este bit se puede descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits enviados es par.
- d) Velocidad de transmisión: Indica el número de bits por segundo que se transfieren, y se mide en baudios (*bauds*). Por ejemplo, 300 baudios representan 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz. Las velocidades de transmisión más comunes para las líneas telefónicas son de 14400, 28800, y 33600. Es posible tener velocidades más altas, pero se reduciría la distancia máxima posible entre los dispositivos.

A continuación tenemos los tipos de conectores de las interfaces seriales generalmente usados con la respectiva indicación de cada pin:

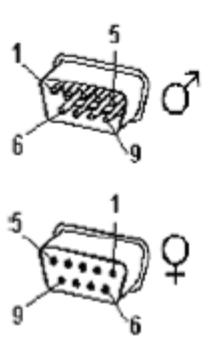
#	Pin	E/S	Función	Conector DB 9
1			Tierra de Chasis	
2	RXD	E	Recibir Datos	
3	TXD	S	Transmitir Datos	
4	DTR	S	Terminal de Datos Listo	
5	SG		Tierra de señal	
6	DSR	E	Equipo de Datos Listo	
7	RTS	S	Solicitud de Envío	
8	CTS	E	Libre para Envío	
9	RI	S	Timbre Telefónico	

Figura 2.19 Pines más importantes del conector DB9

Los estándares de nivel físico son: RS-232, RS-485, Bucle de corriente TTY, RS-422 y Ethernet.

Estándar RS-232:

(Estándar ANSI/EIA-232). Este estándar se utiliza para la comunicación básica serial, trabaja con el método referencial, es decir, necesita un punto de referencia, generalmente es referenciado a tierra. Físicamente no permite conexión a red.

A través de GND pueden acceder picos de voltaje y causar daño en los dispositivos conectados con la interfaz RS 232. La distancia que alcanza sin que se pierda la señal es de 30 metros o 100 pies.⁴⁴

Además trabaja con lógica invertida, es decir, los niveles de voltaje positivos corresponden a 0L⁴⁵ y los niveles de voltaje negativo corresponden a 1L.

La lógica invertida en este caso trabaja:

0L → desde +3 hasta +15 Vcc.

1L → desde -3 hasta -15 Vcc.

Se pueden conectar 2 equipos con par trenzado, transmitiendo a una velocidad de 19,6 kbps y por medio de una comunicación dúplex. El estándar RS 232 define conector del puerto serial.

⁴⁴ Estándar RS2-32, RS4-22, RS4-85, Bucle de corriente TTY, Ethernet.
<http://books.google.com.ec/book?id=xfSJADgd70C&pg=PT307&dq=estandar+422&cd=2#v=onepage&q=bucle%20de%20corriente&f=false>

⁴⁵ 0L: “cero lógico”; 1L: “uno lógico”.

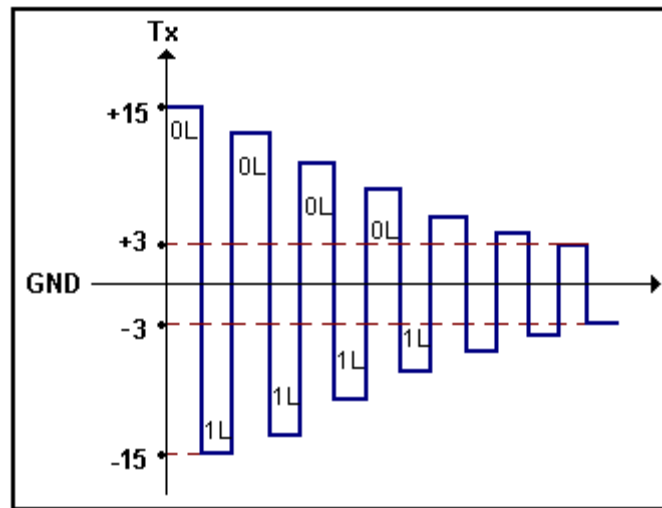


Figura 2.20 RS 232.

Estándar RS 422:

(Estándar EIA RS-422-A). Este estándar necesita 4 hilos conductores. Utiliza el método diferencial para la transmisión de datos, además su comunicación es full dúplex. Alcanza largas distancias de 1,2 a 1,5 kilómetros. En éste método no hay interferencia electromagnética porque los picos de voltaje no influyen en la señal.

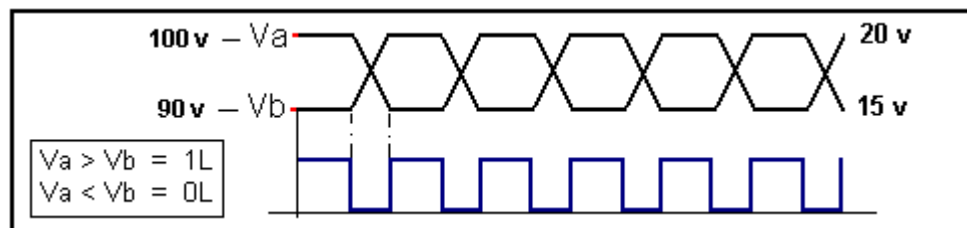


Figura 2.21 RS 422.

La velocidad de transferencia es de 38,4 kbps, su nivel de voltaje es de +1 a +6 voltios. Acepta una red de un maestro y 10 esclavos comunicados por par trenzado.

Bucle de corriente TTY:

Trabaja a una velocidad de transmisión de 16 kbps, alcanza grandes distancias como 1,2 kilómetros, su nivel de corriente es de 0 a 20 mA, su comunicación es dúplex, se pueden conectar sólo dos equipos a través de un cable par trenzado.

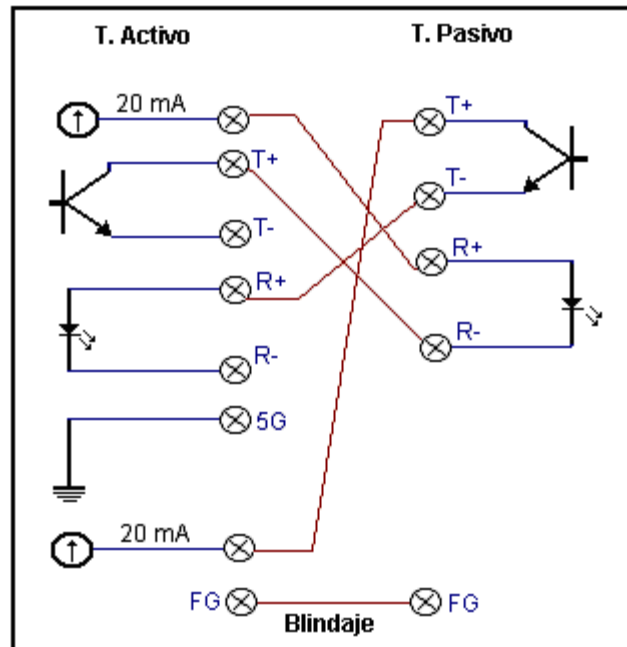


Figura 2.22 Bucle de corriente TTY.

Estándar RS 485:

(Estándar EIA-485). Utiliza el método diferencial, su comunicación es semi dúplex o half dúplex, su red soporta la configuración maestro – esclavo, hasta 32 esclavos. Utiliza 2 hilos, alcanza grandes distancias hasta 4000 pies, su velocidad de transferencia es de 38,4 kbps, su nivel de voltaje es de +1 a +6 voltios.

Este estándar no define conector.

Por ser un método diferencial no hay interferencia electromagnética “sin ruido”, su comunicación en red es por par trenzado.

La resistencia entre la recepción debe ser de 100 a 200 ohmios.

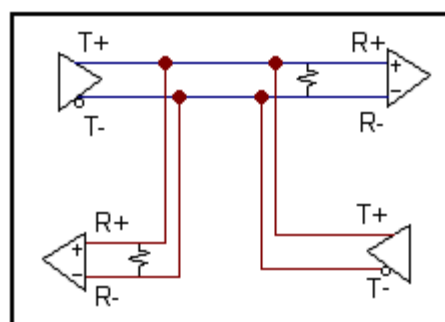


Figura 2.23 RS 485.

Estándar Ethernet:

La velocidad de transferencia es de 10 a 100 Mega bits por segundo, trabaja en modo semi dúplex, es una red eléctrica - óptica, porque transmite por cable y por Fibra Óptica. Alcanza una distancia en red eléctrica de 1,15 kilómetros y en red óptica 200 kilómetros. Cumple con la norma IEEE 802.3 para la industria. Se utiliza para conexión de sistemas entre sí y con PLCs, es un estándar de redes abiertas e industriales a nivel mundial. Funciona con el método de acceso normalizado CSMA/CD. Trabaja con protocolo Map, Tf, Funciones S7, send/recv.

Tiene un elevado rendimiento en la transmisión, trabaja hasta con 1024 estaciones. La topología de la red puede ser en línea, anillo, estrella o árbol.

2.8. INSTRUMENTACIÓN.

2.8.1. DEFINICIONES BÁSICAS.

Binario: Término aplicado a una señal o dispositivo que tiene solo dos posiciones o estados discretos. Cuando es usado en su forma más simple, como en “señal binaria” (lo que es opuesto a señal analógica), el término denota un estado de “encendido-apagado” o de “alto-bajo”.

Controlador: Dispositivo con una salida que varía para regular una variable de control de una manera específica. Un controlador manual varía su salida automáticamente en respuesta a una entrada directa o indirecta de un proceso variable.

Convertidor: Es un dispositivo que recibe información en determinada manera de un instrumento y transmite una señal de salida en otra forma.

Digital: Término aplicado a una señal o dispositivo que usa dígitos binarios para representar valores continuos o estados discretos.

Elemento de control: Dispositivo que controla directamente los valores de la variable manipulada en un lazo de control.

Lazo: Combinación de uno o más instrumentos o funciones de control que señalan el paso de uno a otro con el propósito de medir y/o controlar las variables de un proceso.

Medida: Determinación de la existencia o magnitud de una variable.

Panel: Estructura que tiene un grupo de instrumentos montados sobre ella. El panel puede consistir de una o varias secciones, cubículos, consolas o escritorios.

Variable: Cualquier propiedad variable de un proceso. El término variable de proceso es usado como un estándar para la aplicación a todas las variables.

Controlador lógico programable: Un controlador, usualmente con entradas y salidas múltiples que contiene un programa alterable, es llamado de esta manera o comúnmente conocida como PLC.

Relé: Dispositivo conmutador cuya función es pasar información sin alterarla o solo modificarla en determinada forma.

Sensor: Parte de un lazo o un instrumento que primero detecta el valor de una variable de proceso y que asume una correspondencia, predeterminación, y estado inteligible o salida. El sensor puede ser integrado o separado de un elemento funcional o de un lazo. Al sensor también se le conoce como detector o elemento primario.

Set point: El set point o punto de referencia puede ser establecido manualmente, automáticamente o programado. Su valor se expresa en las mismas unidades que la variable controlada.

Switch: Dispositivo que conecta, desconecta, selecciona, o transfiere uno o más circuitos.

Punto de prueba: Proceso de una conexión el cual no esta permanentemente conectado, su conexión es solamente temporal o intermitente a un instrumento.

Transmisor: Dispositivo que detecta la variable de un proceso a través de un sensor y tiene una salida la cual varía su valor solamente como una función predeterminada de la variable del proceso. El sensor puede estar o no integrado al transmisor.⁴⁶

2.8.2. APLICACIÓN EN LA INDUSTRIA.

La norma es conveniente para el uso en la química, petróleo, generación de poder, aire acondicionado, refinando metales, y otros numerosos procesos industriales. Ciertos campos, como la astronomía, navegación, y medicina, usan instrumentos muy especializados, diferentes a los instrumentos de procesos

⁴⁶

<http://www.hannachile.com/articulos/19/diccionario-conceptos-instrumentacion-procesos.htm>

Diccionario de conceptos básicos de teoría de instrumentación.

industriales convencionales. Se espera que la norma sea flexible, lo bastante para encontrarse muchas de las necesidades de campos especiales.

Aplicación en actividades de trabajo: La norma es conveniente para usar siempre cualquier referencia de un instrumento o de una función de sistema de control se requiere para los propósitos de simbolización e identificación. Pueden requerirse tales referencias para los usos siguientes:

Bocetos del plan, ejemplos instrucción, papeles técnicos, diagramas de sistemas de instrumentación, diagramas de vuelta, diagramas lógicos, descripciones funcionales, diagramas de flujo, dibujos de construcción, especificaciones, órdenes de compra, manifiestos, identificación de instrumentos, funciones de control, instalación, operación e instrucciones de mantenimiento, dibujos y archivos.

2.8.3. SÍMBOLOS DE INSTRUMENTACIÓN.

En instrumentación y control, se emplea un sistema especial de símbolos con el objeto de transmitir de una forma más fácil y específica la información. Esto es indispensable en el diseño, selección, operación y mantenimiento de los sistemas de control.

Un sistema de símbolos ha sido estandarizado por la ISA (Sociedad de Instrumentistas de América). La siguiente información es de la norma: ANSI/ISA-S5.1-1984(R 1992).

Las necesidades de varios usuarios para sus procesos son diferentes. La norma reconoce estas necesidades, proporcionando métodos de simbolismo alternativos. Se mantienen varios ejemplos agregando la información o simplificando el simbolismo, según se desee.

Los símbolos de equipo en el proceso no son parte de esta norma, pero se incluyen para ilustrar aplicaciones de símbolos de la instrumentación.

2.8.3.1. Símbolos y Números de Instrumentación.

La burbuja puede ser usada para etiquetar símbolos distintivos, tal como aquellos para válvulas de control. En otras situaciones la burbuja sirve para representar las propiedades del instrumento.

Un símbolo distintivo cuya relación con el lazo es simplemente aparentar que un diagrama no necesita ser etiquetado individualmente. También, donde hay un elemento primario conectado a otro instrumento en un diagrama, hace uso de un símbolo para representar que el elemento primario en un diagrama puede ser opcional.

Flechas direccionales podrían ser agregadas a las líneas de las señales cuando se necesite aclarar la dirección del flujo para información. Una línea de una señal representara la interconexión entre dos instrumentos en un diagrama de flujo siempre a través de ellos. Pueden ser conectados físicamente por más de una línea.

La secuencia en cada uno de los instrumentos o funciones del lazo está conectada en un diagrama y pueden reflejar el funcionamiento lógico o información acerca del flujo.

Un globo o círculo simboliza a un instrumento aislado o instrumento discreto, para el caso donde el círculo esta dentro de un cuadrado, simboliza un instrumento que comparte un display o un control. Los hexágonos se usan para designar funciones de computadora. Para terminar en los controles lógicos programables PLC's se simbolizan con un rombo dentro de un cuadrado.⁴⁷

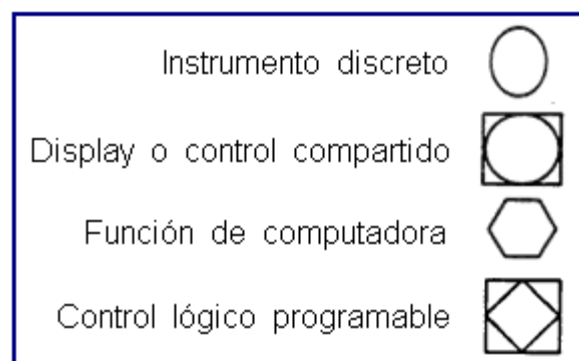


Figura 2.24 Símbolos de instrumentación básicos.

⁴⁷ Cd. Asignatura: Sistemas Scada. Proporcionado por Ing. Carlos Pillajo. Profesor de la carrera de Robótica e IA, de la facultad de Ingeniería en Sistemas de la Universidad Politécnica Salesiana.

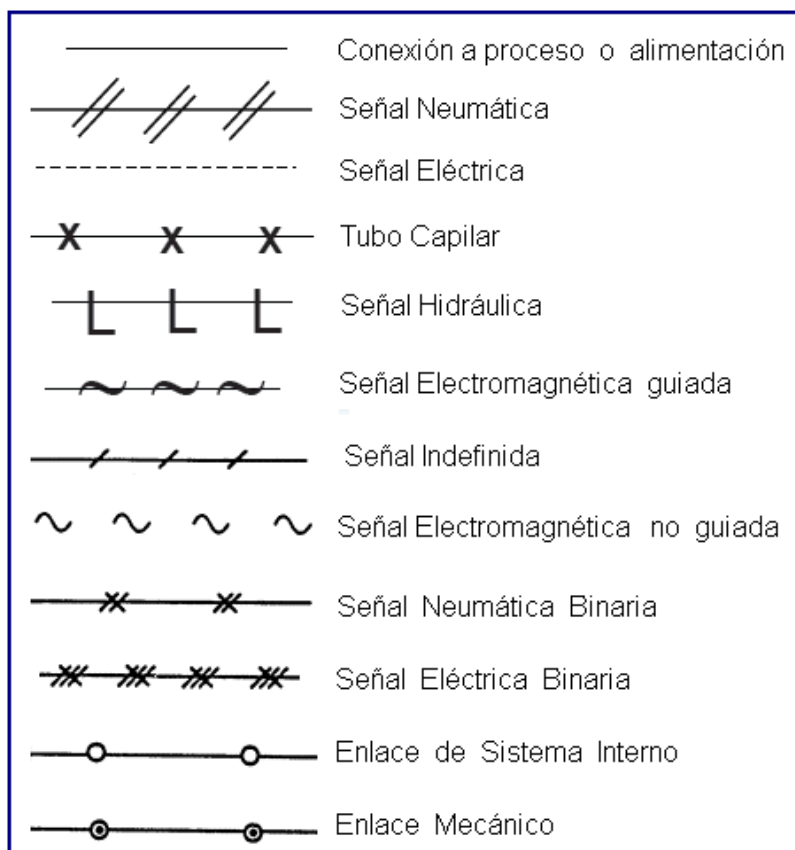


Figura 2.25 Símbolos de líneas de instrumentación.

Los símbolos también indican la posición en que están montados los instrumentos. Los símbolos con o sin líneas nos indican esta información. Las líneas son variadas como son: una sola línea, doble línea o líneas punteadas.

	MONTADO EN TABLERO Normalmente accesible al operador	MONTADO EN CAMPO	UBICACIÓN AUXILIAR Normalmente accesible al operador.
Instrumento Discreto o Aislado			
Display compartido, Control compartido.			
Función de Computadora			
Control Lógico Programable			

Figura 2.26 Posición de montaje de los instrumentos.

Las líneas punteadas indican que el instrumento esta montado en la parte posterior del panel el cual no es accesible al operador:

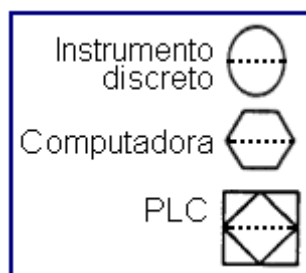


Figura 2.27 Montaje posterior del panel.

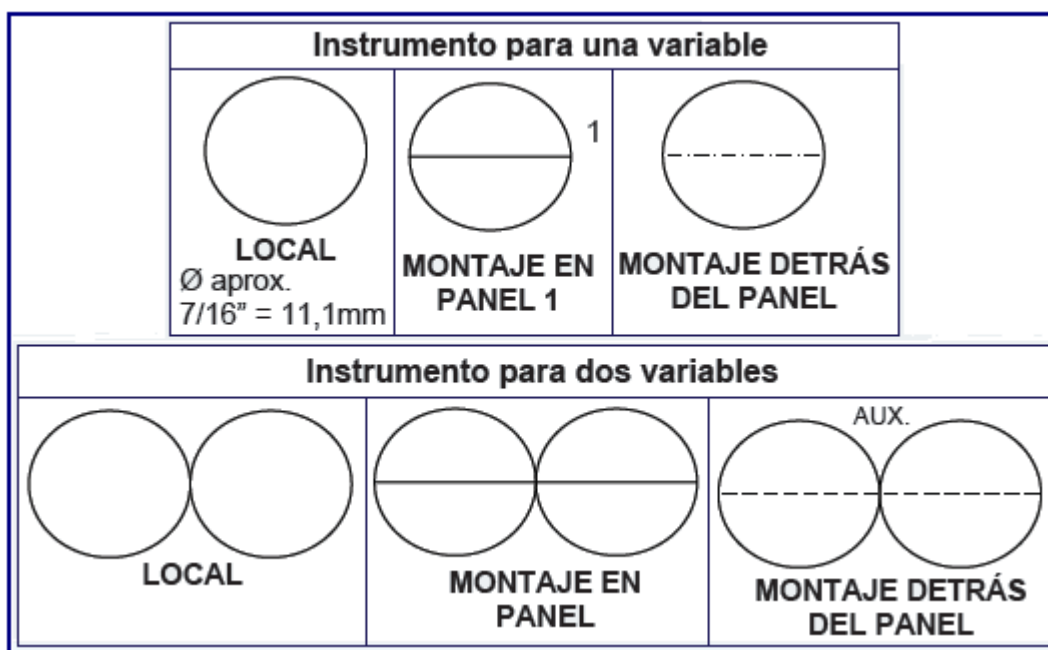


Figura 2.28 Simbología general.⁴⁸

Numero de identificación de los instrumentos o números TAG:

Cada instrumento o función para ser designada esta diseñada por un código alfanumérico o etiquetas con números. La parte de identificación del lazo del número de etiqueta generalmente es común a todos los instrumentos o funciones del lazo. Un sufijo o prefijo puede ser agregado para completar la identificación.

El número de lazo del instrumento puede incluir información codificada, tal como la designación del área de la planta que lo designe. Esto también es posible para series específicas de números para designar funciones especiales. Cada

⁴⁸ <http://www.geocities.com/roccotarantino23/Blogmediciondevariables1.pdf>
instrumentación industrial.

instrumento puede ser representado en diagramas por un símbolo. El símbolo puede ser acompañado por un número de etiqueta.

NUMERO DE IDENTIFICACION TIPICO (NUMERO TAG)	
TIC 103	Identificación del instrumento o número de etiqueta
T 103	Identificación de lazo
103	Número de lazo
TIC	Identificación de funciones
T	Primera letra
IC	Letras Sucesivas
NUMERO DE IDENTIFICACION EXPANDIDO	
10-PAH-5A	Número de etiqueta
10	Prefijo opcional
A	Sufijo opcional
Nota: Los guiones son optativos como separadores.	

Figura 2.29 Números de Identificación.

2.8.4. IDENTIFICACIÓN FUNCIONAL.

La identificación funcional de un instrumento o su equivalente funcional consiste de letras, las cuales se muestran en la Figura 2.30 que incluyen una primera letra (designación de la medida o variable inicial) y una o más letras sucesivas (identificación de funciones).

La identificación funcional de un instrumento esta hecha de acuerdo a su función y no a su construcción.

Un registrador de diferencia de presión usado para medir flujo se identifica como *FR*; un indicador de presión y un switch actuado a presión conectado a la salida de un transmisor de nivel neumático están identificados por *LI* y *LS*, respectivamente.

En un lazo de instrumentos, la primera letra de una identificación funcional es seleccionada de acuerdo a la medida y a la variable inicial y no de acuerdo a la variable manipulada. Una válvula de control varía el flujo de acuerdo a lo dictaminado por un controlador de nivel, esto es una *LV*.

1° Letra		2° Letra		
Variable medida(3)	Letra de Modificación	Función de lectura pasiva	Función de Salida	Letra de Modificación
A. Análisis (4)		Alarma		
B. Llama (quemador)		Libre (1)	Libre (1)	Libre (1)
C. Conductividad			Control	
D. Densidad o Peso específico	Diferencial (3)			
E. Tensión (Fem.)		Elemento Primario		
F. Caudal	Relación (3)			
G. Calibre		Vidrio (8)		
H. Manual				Alto (6)(13)(14)
I. Corriente Eléctrica		Indicación o indicador		
J. Potencia	Exploración (6)			
K. Tiempo			Estación de Control	
L. Nivel		Luz Piloto (10)		Bajo (6)(13)(14)
M. Humedad				Medio o intermedio
N. Libre(1)		Libre	Libre	Libre
O. Libre(1)		Orificio		
P. Presión o vacío		Punto de prueba		
Q. Cantidad	Integración (3)			
R. Radiactividad		Registro		
S. Velocidad o frecuencia	Seguridad (7)		Interruptor	
T. Temperatura			Transmisión o transmisor	
U. Multivariable (5)		Multifunción (11)	Multifunción (11)	Multifunción (11)
V. Viscosidad			Válvula	
W. Peso o Fuerza		Vaina		
X. Sin clasificar (2)		Sin clasificar	Sin clasificar	Sin clasificar
Y. Libre(1)			Relé o compensador	Sin clasificar
Z. Posición			Elemento final sin clasificar	

Figura 2.30 Identificación Funcional.

Ejemplos:

LV: Válvula de nivel.

TDAL: Temperatura diferencial alarma baja.

FR: Registrador de caudal o flujo.

PCV: Válvula controladora de presión.

Se sugieren las siguientes abreviaturas para representar el tipo de alimentación:

AS: Alimentación de aire, **ES:** Alimentación eléctrica, **GS:** Alimentación de gas,

HS: Alimentación hidráulica, **NS:** Alimentación de nitrógeno, **SS:** Alimentación de

vapor, **WS:** Alimentación de agua.

2.8.5. IDENTIFICACIÓN Y DIAGRAMA DEL LAZO.

La identificación del lazo consiste en la primera letra y un número. Cada instrumento en un lazo tiene asignado a él el mismo número de lazo y, en caso de una numeración paralela, la misma primera letra. Cada lazo de instrumentos tiene un único número de identificación de lazo. Un instrumento común a dos o más lazos podría cargar la identificación del lazo al cual se le considere predominante. La numeración de los lazos puede ser paralela o serial. En paralela por ejemplo: TIC-100, FRC-100, LIC-100, AL-100, en serial por ejemplo: 001, 301 o 1201. También se pueden añadir sufijos como A, B, C, etc.

Los diagramas de instrumentación del proceso o diagramas de instrumentación y tuberías, son una buena fuente de información incluyendo todas las variables del proceso en el sistema como también la información de cada uno de los instrumentos en los lazos. Cuando se necesita más información específica y detallada es necesario utilizar otros tipos de diagramas. El diagrama de lazo nos permite una mejor comprensión de cómo opera el lazo. Esta información nos permite identificar las conexiones entre los dispositivos, la acción de los componentes y las rutas de comunicación.

Un ejemplo a continuación:

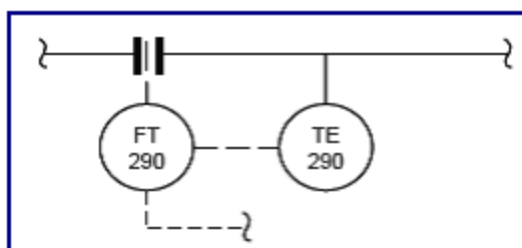


Figura 2.31 Ejemplo de un diagrama de lazo.

Montaje en campo: Transmisor de caudal o flujo en lazo 290 conectado eléctricamente a un elemento primario o sensor de temperatura en lazo 290.

2.8.5.1. Set Point y Rango de Operación.

Por medio del diagrama de lazo de instrumentación se puede identificar el punto de ajuste o Set Point y el rango de operación de los instrumentos.

Para la información acerca del rango de operación del instrumento se localiza dentro de un rectángulo horizontal cerca del instrumento, al que se está haciendo

referencia, mientras que la información acerca del punto de ajuste o set point se muestra en un rombo ubicado cerca del controlador.

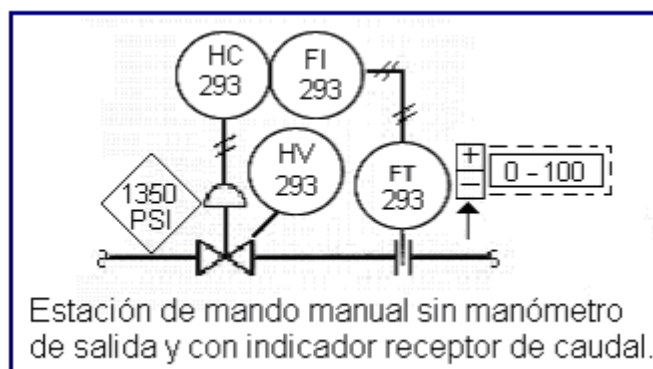


Figura 2.32 Set point y rango de operación.

2.8.6. LAZOS ELECTRÓNICOS.

Una de las ventajas de los diagramas de lazo es el poder hacer un seguimiento lógico a través del lazo. Por lo regular el punto de inicio para leer un diagrama es por el lado izquierdo, el propósito es poder encontrar el elemento primario.

El siguiente ejemplo se refiere a un lazo diseñado para la medición del flujo por medio de una placa de orificio. El transmisor FT-101 mide la diferencia de presión causada por la restricción de la placa de orificio.

El transmisor también produce una señal que representa esta caída, la cual es proporcional al flujo promedio. Los puntos de conexión del transmisor son terminales eléctricos. Los signos positivo y negativo indican la polaridad de las terminales. Este transmisor emite una señal de 4 a 20 mA. La flecha apuntando hacia arriba nos indica que es un instrumento que actúa directamente. El ovalo alrededor de las líneas de señal indica que esta blindada la señal para evitar interferencia eléctrica que pueda ocasionar una lectura errónea en los indicadores.

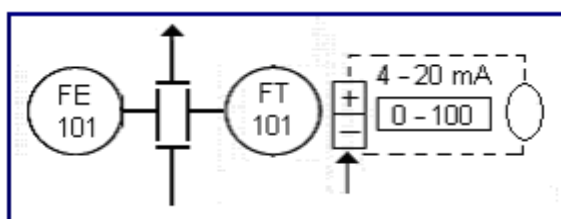


Figura 2.33 Ejemplo de un lazo electrónico.

También existen lazos neumáticos, hidráulicos y otros, se clasifican por lo general de acuerdo a la señal principal que engloba el sistema.

A continuación se presenta el diagrama de lazo de instrumentación del proyecto:

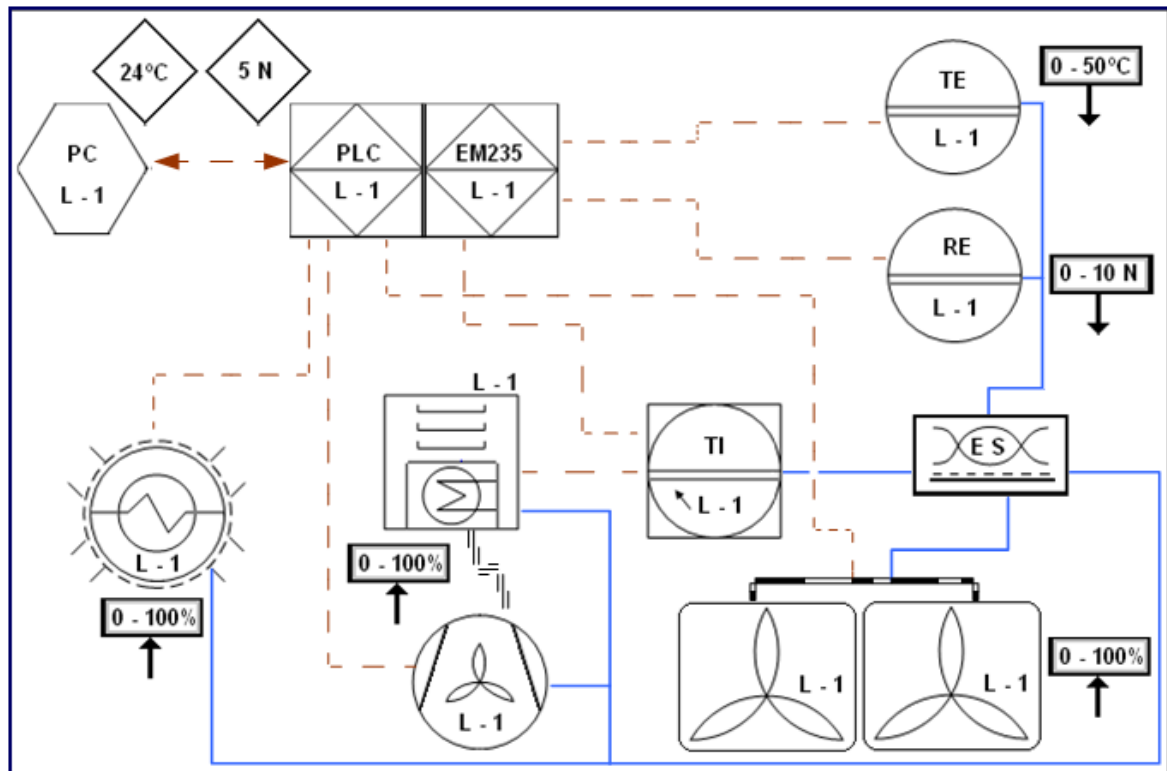


Figura 2.34 Diagrama de instrumentación del proyecto de tesis.

Para empezar todos los instrumentos se establecen en Lazo 1 (L – 1), ya que empiezan a trabajar al mismo ciclo.


Describiendo por la esquina superior izquierda, tenemos: Función de Computadora, conectada bidireccionalmente con el Controlador Lógico Programable (PLC y el módulo EM235), por medio de una interfaz serial; se asignan los Set Points o puntos de ajuste a 24°C correspondientes a la temperatura y a 5N (5º nivel de luminosidad) correspondientes a la luz.

Conectado al módulo por medio eléctrico, tenemos 2 sensores, el primero de temperatura (TE), con un rango de operación de 0 a 50°C, la flecha hacia abajo nos indica que la entrada (Temperatura) es inversamente proporcional a la salida (Resistencia) entregada por el sensor (NTC). El segundo sensor es el de luz (RE), con un rango de operación de 0 a 10 N (niveles de luminosidad), la flecha

hacia abajo nos indica que la entrada (Radiación de Luz) es inversamente proporcional a la salida (Resistencia) entregada por el sensor (Fotorresistencia).

Los sensores y actuadores están alimentados por una fuente de corriente eléctrica (ES) guiada por la línea en celeste.

En la esquina inferior derecha, tenemos 2 ventiladores conectados en paralelo y comunicados eléctricamente al PLC, con un rango de operación de 0 al 100%.

El TI es un Indicador de datos en general como: Temperatura, Luz y Calefacción. Luego tenemos un acople del calefactor (niquelina) con un ventilador, para crear la calefacción por convección, por medio de un conducto de aire:  este acople trabaja en un rango de 0 a 100%.

Por último en la parte inferior izquierda, tenemos el actuador Lámpara, que opera en un rango de 0 a 100% de su acción y que está conectada eléctricamente al PLC.

CAPÍTULO 3. HARDWARE

En el presente capítulo, se explican todos y cada uno de los dispositivos utilizados en el proyecto como son los sensores, actuadores, PLC, módulo de extensión analógico y el prototipo de prueba que consta de instrumentos electrónicos.

3.1. SENSORES.

Los sensores son la principal herramienta para la obtención de los datos del entorno del sistema. Un sensor es un aparato capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas.

Las variables de instrumentación dependen del tipo de sensor y pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc.

Una magnitud eléctrica obtenida puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como un fototransistor), etc.⁴⁹

Puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable a medir o a controlar, mientras que un transductor está compuesto de un circuito de acondicionamiento de la señal que entrega el sensor en bruto.

Hay sensores que no solo sirven para medir la variable, sino también para convertirla mediante circuitos electrónicos en una señal estándar (4 a 20 mA, o 1 a 5Vdc) para tener una relación lineal con los cambios de la variable sensada dentro de un rango, para fines de control de dicha variable en un proceso.

⁴⁹ www.elprisma.com/apuntes/ingenieria_industrial/transductoressensores/

Por lo general, la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento, como por ejemplo un puente de Wheatstone⁵⁰, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para el resto de la circuitería.

3.1.1. ÁREAS DE APLICACIÓN DE LOS SENSORES.

Los sensores se utilizan en la industria automotriz, en la industria aeroespacial, en la medicina, en la industria de manufactura, en Robótica⁵¹, en el campo de seguridad, en Domótica⁵², en la automatización de industrias y de sistemas en general, en sistemas HMI/Scada, etc.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor, etc.

3.1.2. CARACTERÍSTICAS DE UN SENSOR.

Entre las características técnicas de un sensor destacan las siguientes:

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: es el error de medida máximo esperado.
- *Offset* o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el *offset*.
- Linealidad o correlación lineal.
- Sensibilidad de un sensor: relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada.

⁵⁰ WHEATSTONE: Un puente de Wheatstone es un instrumento eléctrico de medida que se utiliza para medir resistencias desconocidas mediante el equilibrio de los brazos del puente, éstos están constituidos por cuatro resistencias que forman un circuito cerrado, siendo una de ellas la resistencia bajo medida.

http://fisica.udea.edu.co/~lab-gicm/2009_electronica/2009_Puente_de_Wheatstone.pdf

⁵¹ ROBÓTICA: Ciencia y Tecnología de los robots. Se ocupa del diseño, manufactura y aplicaciones de los robots. La robótica combina diversas disciplinas como son: mecánica, electrónica, informática, inteligencia artificial e ingeniería de control. [www.industria.udc.es/Academicos/AlexanderBorger/Docts_Docencia/Seminario de Aut/trabajos/Trabajos_2005/Robotica/4.-LA_ROBÓTICA.htm](http://www.industria.udc.es/Academicos/AlexanderBorger/Docts_Docencia/Seminario_de_Aut/trabajos/Trabajos_2005/Robotica/4.-LA_ROBÓTICA.htm)

⁵² DOMÓTICA: Domus significa *casa* en latín y *tica* significa *automática* en griego. Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda. www.mtm.es/proyec/domotica/domt_def.htm

- Resolución: mínima variación de la magnitud de entrada que puede apreciarse a la salida.
- Rapidez de respuesta: puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo: oxidación, desgaste, envejecimiento, etc.
- Repetitividad: error esperado al repetir varias veces la misma medida.

3.1.3. RESOLUCIÓN Y PRECISIÓN.

La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida. Sin embargo, la precisión es el máximo error esperado en la medida.

La resolución puede ser de menor valor que la precisión. Por ejemplo, si al medir una distancia la resolución es de 0,01 mm, pero la precisión es de 1 mm, entonces pueden apreciarse variaciones en la distancia medida de 0,01 mm, pero no puede asegurarse que haya un error de medición menor a 1 mm. Si el error en la medida sigue una distribución normal o similar, lo cual es frecuente en errores accidentales, la repetitividad podría ser de un valor inferior a la precisión.

Sin embargo, la precisión no puede ser de un valor inferior a la resolución, pues no puede asegurarse que el error en la medida sea menor a la mínima variación en la magnitud de entrada que puede observarse en la magnitud de salida.

3.1.4. TIPOS DE SENSORES.

En la siguiente tabla se indican algunos tipos y ejemplos de sensores electrónicos:

MAGNITUD	TRANSDUCTOR	TIPO
Posición lineal o angular	Potenciómetro	Analógica
	Encoder	Digital
Desplazamiento y deformación	Transformador de variación lineal	Analógica
	Galga extensiométrica	Analógica
	Magnetostrictivos	A/D
	Magnetorresistivos	Analógica
	LVDT	Analógica
Velocidad lineal y angular	Dinamo tacométrica	Analógica
	Encoder	Digital
	Detector inductivo	Digital
	Servo-inclinómetros	A/D
	RVDT	Analógica
	Giróscopo	
Aceleración	Acelerómetro	Analógico
	Servo-accelerómetros	
Fuerza y par (deformación)	Galga extensiométrica	Analógico
	Triaxiales	A/D
Presión	Membranas	Analógica
	Piezoeléctricos	Analógica
	Manómetros Digitales	Digital
Caudal	Turbina	Analógica
	Magnético	Analógica
Temperatura	Termopar	Analógica
	RTD	Analógica
	Termistor NTC	Analógica
	Termistor PTC	Analógica
	Bimetal	I/O
Sensores de presencia	Inductivos	I/O
	Capacitivos	I/O
	Ópticos	I/O y Analógica
Sensores táctiles	Matriz de contactos	I/O
	Piel artificial	Analógica
Visión artificial	Cámaras de video	Proceso digital
	Cámaras CCD o CMOS	Proceso digital
Sensor de proximidad	Sensor final de carrera	
	Sensor capacitivo	
	Sensor inductivo	
	Sensor fotoeléctrico	
Sensor acústico	micrófono	
Sensores de acidez	IsFET	
Sensor de luz	fotodiodo	
	Fotorresistencia	
	Fototransistor	
	Célula fotoeléctrica	
Sensores de movimiento	Sensores inerciales	

Figura 3.1 Tipos de sensores.

El presente proyecto utiliza 2 sensores: sensor de temperatura mediante un Termistor NTC y un sensor de luz con una Fotorresistencia.

3.1.5. SENSOR DE TEMPERATURA.

Un termistor es un semiconductor que varía el valor de su resistencia eléctrica en función de la temperatura, su nombre proviene de Thermally sensitive resistor (Resistor sensible a la temperatura en inglés). Existen dos clases de termistores: NTC y PTC.⁵³

Tipo NTC.- Negative Temperature Coefficient (Coeficiente Negativo de Temperatura en Inglés), es una resistencia variable cuyo valor va decreciendo a medida que aumenta la temperatura. Son resistencias constituidas por un cuerpo semiconductor. Se emplean en su fabricación óxidos semiconductores de níquel, zinc, cobalto, etc.

La relación entre la resistencia y la temperatura no es lineal sino exponencial.

La característica tensión-intensidad (V/I) de un termistor NTC presenta un carácter peculiar ya que, cuando las corrientes que lo atraviesan son pequeñas, el consumo de potencia es demasiado pequeño para registrar incrementos apreciables de temperatura o descensos en su resistencia; en esta parte de la característica, la relación tensión-intensidad será prácticamente lineal y en consecuencia cumplirá la ley de Ohm.

Si aumenta la tensión aplicada al termistor, se llegará a un valor de intensidad en que la potencia consumida provocará aumentos de temperatura suficientemente grandes como para que la resistencia del termistor NTC disminuya apreciablemente, incrementándose la intensidad hasta que se establezca el equilibrio térmico.

Tipo PTC Positive Temperature Coefficient (Coeficiente positivo de temperatura), es una resistencia variable cuyo valor aumenta a medida que se incrementa la temperatura.

Los termistores PTC se utilizan en una gran variedad de aplicaciones: limitación de corriente, sensor de temperatura, desmagnetización y para la protección contra el recalentamiento de equipos tales como motores eléctricos. También se utilizan en indicadores de nivel, para provocar retardos en circuitos, como termostatos y como resistores de compensación.

⁵³

www.gii.upv.es/personal/gbenet/treballs%20cursos%20anteriors-TIM-IIN-INYP-AYPD/termistores/termistores.doc

El termistor PTC pierde sus propiedades y puede comportarse eventualmente de una forma similar al termistor NTC si la temperatura llega a ser demasiado alta.

Las aplicaciones de un termistor PTC están, restringidas a un determinado margen de temperaturas.

Hasta un determinado valor de voltaje, la característica I/V sigue la ley de Ohm, pero la resistencia aumenta cuando la corriente que pasa por el termistor PTC provoca un calentamiento y se alcanza la temperatura de conmutación. La característica I/V depende de la temperatura ambiente y del coeficiente de transferencia de calor con respecto a dicha temperatura.

En el proyecto se utiliza un termistor NTC que sensa la temperatura del refrigerante en el motor de un automóvil, posee un rango de medida desde -40°C hasta $+130^{\circ}\text{C}$. Aunque el rango que se necesita es de sólo 0°C a 50°C .

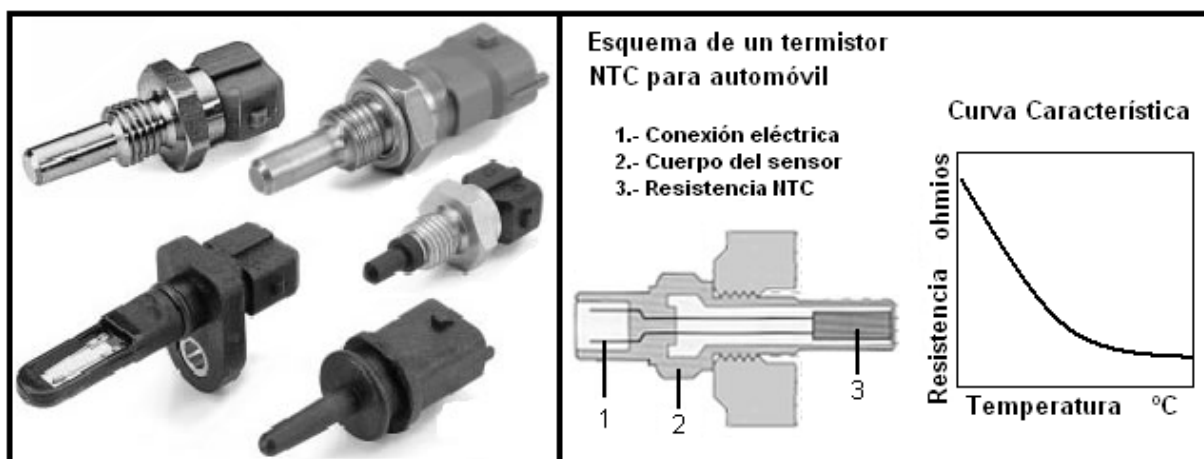


Figura 3.2 Termistor NTC de automóvil.⁵⁴

La curva del termistor utilizado se la obtuvo experimentalmente, midiendo la resistencia en cada cambio de temperatura:

⁵⁴ <http://www.mecanicavirtual.org/sensores7.htm>

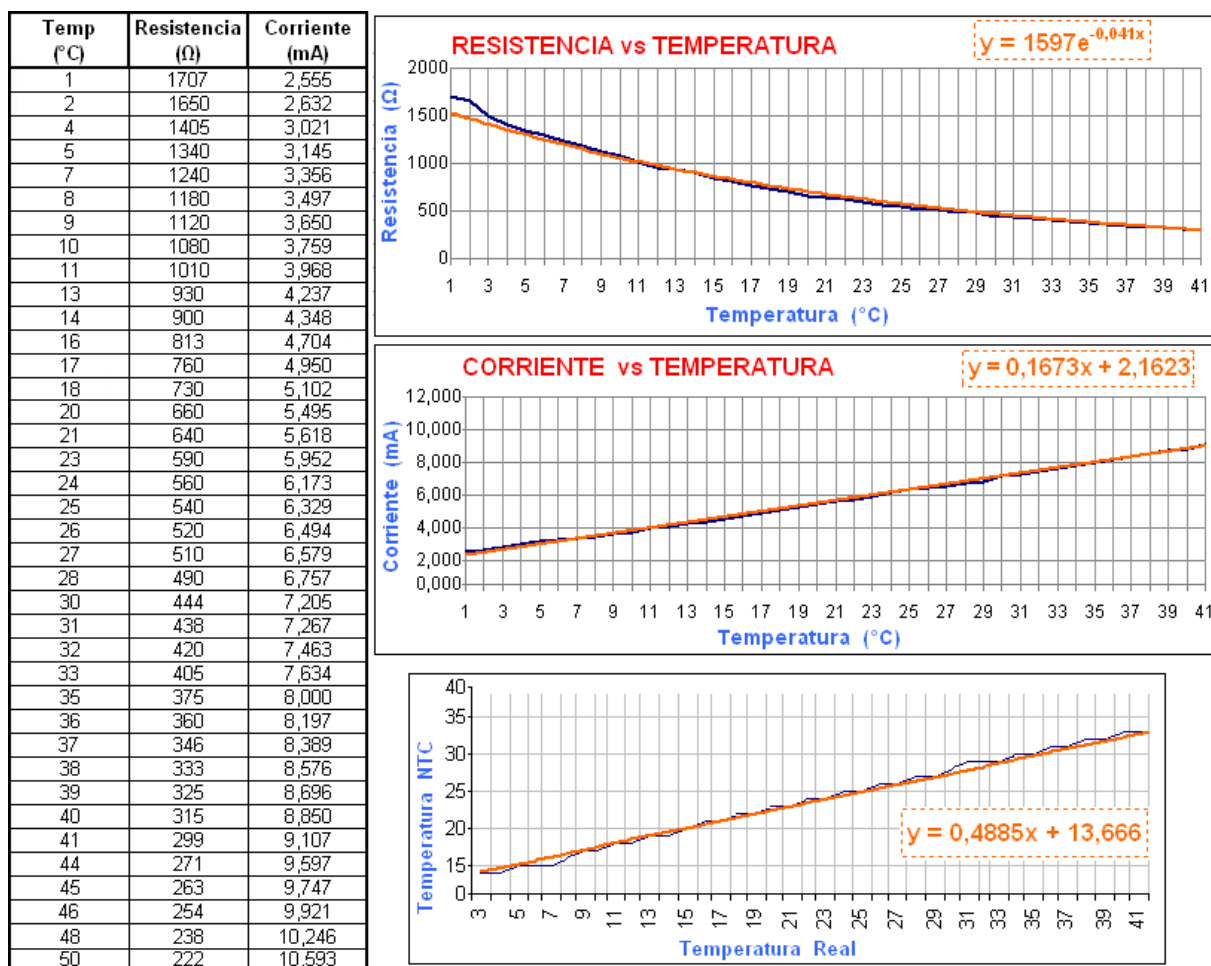


Figura 3.3 Curvas R/T, I/T, Temperatura Real del termistor NTC.

La configuración del sensor de temperatura con termistor NTC es:

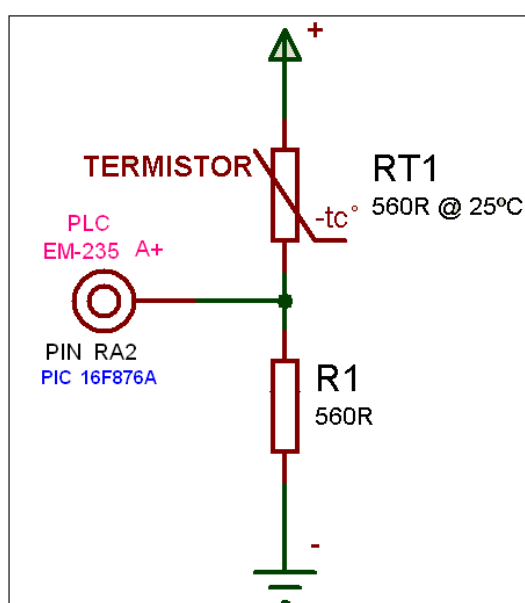


Figura 3.4 Diagrama del sensor de temperatura.

3.1.6. SENSOR DE LUZ.

Se utiliza como sensor de luz, una fotorresistencia.

Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotoresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz. Sus siglas: LDR, se originan de su nombre en inglés *light-dependent resistor* (Resistencia dependiente de la luz). Su cuerpo está formado por una célula o celda y dos patillas.

El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (puede descender hasta 50 ohms) y muy alto cuando está a oscuras (varios megaohms).

Un fotorresistor está hecho de un semiconductor de alta resistencia como el sulfuro de cadmio, CdS. Si la luz que incide en el dispositivo es de alta frecuencia, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar la banda de conducción. El electrón libre que resulta, y su hueco asociado, conducen la electricidad, de tal modo que disminuye la resistencia. Las células de sulfuro del cadmio se basan en la capacidad del cadmio de variar su resistencia según la cantidad de luz que incide la célula. Cuanta más luz incide, más baja es la resistencia.⁵⁵

Las celdas son también capaces de reaccionar a una amplia gama de frecuencias, incluyendo infrarrojo (IR), luz visible, y ultravioleta (UV).

La variación del valor de la resistencia tiene cierto retardo, diferente si se pasa de oscuro a iluminado o de iluminado a oscuro. Esto limita a no usar los LDR en aplicaciones en las que la señal luminosa varía con rapidez. El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo. Esta lentitud da ventaja en algunas aplicaciones, ya que se filtran variaciones rápidas de iluminación que podrían hacer inestable un sensor. Se fabrican en diversos tipos y pueden encontrarse en muchos artículos de consumo, como por ejemplo en cámaras, medidores de luz, relojes con radio, alarmas de seguridad o sistemas de encendido y apagado del alumbrado de calles.

También se fabrican fotoconductores de Ge-Cu que funcionan dentro de la gama más baja "radiación infrarroja".

⁵⁵ <http://www.sabelotodo.org/electrotecnia/electrosensores.html>

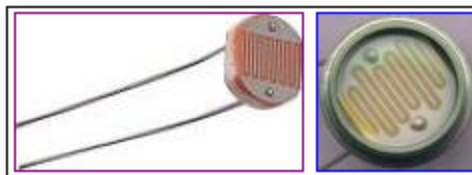


Figura 3.5 Imagen Fotorresistencia.

La curva de la fotorresistencia, se la obtuvo experimentalmente, midiendo la resistencia en cada cambio de nivel de luz a través de la variación de voltaje de un foco dicróico:

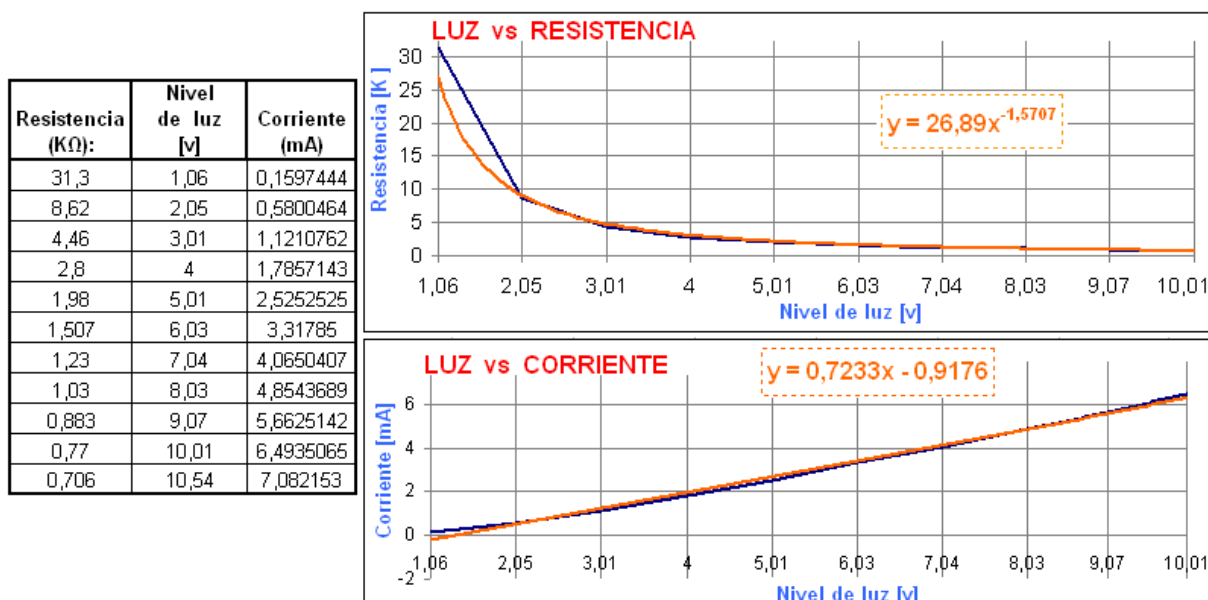


Figura 3.6 Curva R/L, I/L.

La configuración del sensor de luz con fotorresistencia es:

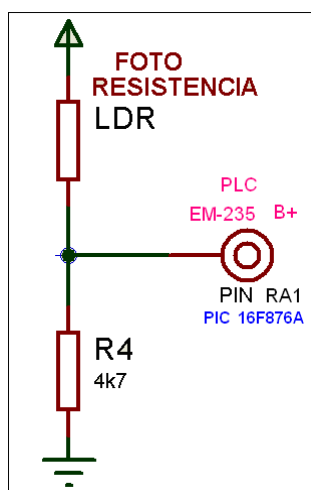


Figura 3.7 Diagrama del sensor de luz con Fotorresistencia.

3.2. ACTUADORES.

Los actuadores son dispositivos capaces de generar una fuerza a partir de líquidos, de energía eléctrica o gaseosa. El actuador recibe la orden de un regulador o controlador y da una salida necesaria para activar a un elemento final de control como lo son: las válvulas, los motores, las niquelinas, las lámparas, etc. Existen tres tipos de actuadores: Hidráulicos, Neumáticos y Eléctricos.

Los actuadores hidráulicos se emplean cuando lo que se necesita es potencia y se basan en la presión de líquidos como por ejemplo aceites hidráulicos ya que soportan altas presiones.

Los actuadores neumáticos son simples posicionamientos, y se basan en la presión de aire comprimido.

Los actuadores eléctricos son muy utilizados en los aparatos mecatrónicos, como por ejemplo en los robots; su funcionamiento se basa en la alimentación eléctrica.

Los hidráulicos requieren demasiado equipo para suministro de energía, así como de mantenimiento periódico, las aplicaciones de los actuadores neumáticos son limitadas desde el punto de vista de precisión y mantenimiento.⁵⁶

Por todo esto es necesario conocer muy bien las características de cada actuador para utilizarlos correctamente de acuerdo a la aplicación específica, por consiguiente los actuadores utilizados en el proyecto son eléctricos y constituyen: Ventiladores, Calefactor y Lámpara.

3.2.1. VENTILADORES.

Un ventilador es una máquina de fluido concebida para producir una corriente de aire mediante un rodete con aspas que giran produciendo una diferencia de presiones. Entre sus aplicaciones, destacan las de hacer circular y renovar el aire en un lugar cerrado para proporcionar oxígeno suficiente a los ocupantes y eliminar olores, principalmente en lugares cerrados; así como la de disminuir la resistencia de transmisión de calor por convección.

Los ventiladores se utilizan para desplazar aire o gas de un lugar a otro, dentro de espacios, para motivos industriales o uso residencial, para ventilación o para

⁵⁶ <http://www.tvtronica.com.ar/Actuadores.htm>

aumentar la circulación de aire en un espacio habitado, básicamente para refrescar. Por esta razón es un elemento indispensable en climas cálidos.⁵⁷

En el proyecto se utilizan ventiladores de 12V de corriente continua, típicos de las computadoras de escritorio.



Figura 3.8 Ventilador.

3.2.2. CALEFACTOR.

El calefactor es un aparato, normalmente eléctrico, que proporciona a una estancia o recipiente un flujo rápido de aire caliente continuo mediante un radiador que genera una fuente de calor y un ventilador que calienta rápidamente el aire y lo transmite al lugar en que se encuentre.

Un calefactor eléctrico es un dispositivo que produce energía calorífica a partir de la eléctrica. El tipo más difundido es el calefactor eléctrico "resistivo", donde la generación del calor se debe al Efecto Joule⁵⁸.

Otros calefactores eléctricos menos conocidos son los "termoeléctricos", que intercambian calor mediante un sistema más complicado: el Efecto Peltier⁵⁹.

Entre las aplicaciones más conocidas del efecto Joule se tienen los elementos de las estufas para calentar el ambiente, los filamentos de los secadores para el pelo, las resistencias de las planchas para la ropa, las hornallas o fogones de las cocinas, las resistencias de tostadores y hornos industriales, los calentadores en

⁵⁷ www.eccpn.aibarra.org/temario/seccion5/capitulo82/capitulo82.htm

⁵⁸ EFECTO JOULE.- Si en un conductor circula corriente eléctrica, parte de la energía cinética de los electrones se transforma en calor debido a los choques que sufren con los átomos del material conductor por el que circulan, elevando la temperatura del mismo. [http://www.unavarra.es/ets02/Introduccion%20a%20TE\(c\).htm](http://www.unavarra.es/ets02/Introduccion%20a%20TE(c).htm)

⁵⁹ EFECTO PELTIER.- Creación de diferencia térmica a partir de una diferencia de potencial eléctrico, cuando una corriente pasa a través de dos metales diferentes (tipo-n y tipo-p) que están conectados entre sí en dos soldaduras (uniones Peltier). <http://www.profisica.cl/comofuncionan/como.php?id=15>

los hervidores de agua y fermentadores, los alambres para evitar el congelamiento en refrigeradores y el empañamiento en vidrios de las ventanas traseras de automóviles, los calefactores en peceras e invernaderos, etc.

Para evitar que se funda el conductor hay que transferir el calor generado por efecto Joule, para mejorar esa transmisión térmica, en general los calefactores tienen mayor área o superficie de contacto con el medio que les rodea. Dependiendo de la aplicación, el calor se transfiere en una o más de las 3 formas posibles:

- Por conducción (hervidores, planchas, desempañadores, etc.)
- Por convección (secadores de pelo, calentadores de aire, etc.)
- Por radiación (tostadores, estufas de cuarzo, etc.)

En el presente proyecto se utiliza como calefactor, una niquelina de secador de cabello con el principio de la convección.



Figura 3.9 Imágenes de calefactores por convección.

3.2.3. LÁMPARA.

Las luminarias (conocidas como lámparas) son aparatos que sirven de soporte y conexión a la red eléctrica con el objetivo de aportar luz. Como esto no basta para que cumplan eficientemente su función, es necesario que cumplan una serie de características ópticas, mecánicas y eléctricas entre otras.⁶⁰

A nivel de óptica, la luminaria es responsable del control y la distribución de la luz emitida por la lámpara. Es importante que en el diseño de su sistema óptico se cuide la forma y distribución de la luz, el rendimiento de las luminarias y el

⁶⁰

http://books.google.com.ec/books?id=W8EnsQ2_q0cC&pg=PT423&dq=luminarias&cd=1#v=onepage&q=luminarias&f=false Luminarias.

deslumbramiento que pueda provocar en los usuarios. Otros requisitos que deben cumplir las luminarias es que sean de fácil instalación y mantenimiento. Para ello, los materiales empleados en su construcción han de ser los adecuados para resistir el ambiente en que deba trabajar la luminaria y mantener la temperatura de la lámpara dentro de los límites de funcionamiento. Todo esto sin perder de vista aspectos no menos importantes como la economía o la estética.

En el trabajo presente se emplea como lámpara, un foco dicroico halógeno de 50 watts a 12 voltios.



Figura 3.10 Imágenes de lámparas y focos halógenos dicroicos.⁶¹

3.3. DIAGRAMAS.

Se han realizado algunos tipos de diagramas como: de componentes y distribución, de casos de uso, de estados, de secuencia; con el fin de lograr mayor entendimiento del hardware y su conexión.

3.3.1. DIAGRAMA DE COMPONENTES Y DISTRIBUCIÓN.

Un componente es una parte física de un sistema, y se encuentra en la computadora, no en la mente del analista. Ejemplos de componentes son tablas, archivos de datos, ejecutables, bibliotecas de vínculos dinámicos, documentos, ventanas, etc.⁶²

Lo que contiene este diagrama son interfaces y relaciones, aunque también pueden aparecer otros tipos de símbolos. El símbolo principal de un diagrama de componentes es un rectángulo que tiene otros dos sobrepuestos en su lado izquierdo, con el nombre del componente dentro del rectángulo más grande por ejemplo:

⁶¹ <http://images.google.com.ec/images?hl=es&um=1&sa=1&q=im%C3%A1genes+dicroico&aq=f&oq=&start=0>

⁶² <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>



Figura 3.11 Ejemplo de un componente.

Las relaciones de dependencia entre componentes se dibujan con líneas entrecortadas y flechas. Los diagramas de distribución se enfocan específicamente al hardware de un sistema determinado.

El elemento primordial del hardware es un *nodo*, que es un nombre genérico para todo tipo de recurso de cómputo y se representa mediante un cubo, dentro del cubo se puede introducir información sobre el nodo, que puede ser simplemente texto o inclusive componentes, usando los diagramas de componentes anteriormente señalados.

Los diagramas de distribución se encargan de dibujar hardware, pero se los puede detallar con los diagramas de componentes en su interior, cada uno de los nodos puede contener otros componentes, incluyendo software, lo cual debe ser especificado.

Es decir se pueden combinar los diagramas de Componentes y de Distribución.

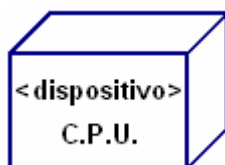


Figura 3.12 Ejemplo de un nodo.

En el presente proyecto se realizan dos diagramas de distribución y componentes, el primero estableciendo que el proceso de lógica difusa se realiza en LabView y el segundo desarrollando lógica difusa en Step7:

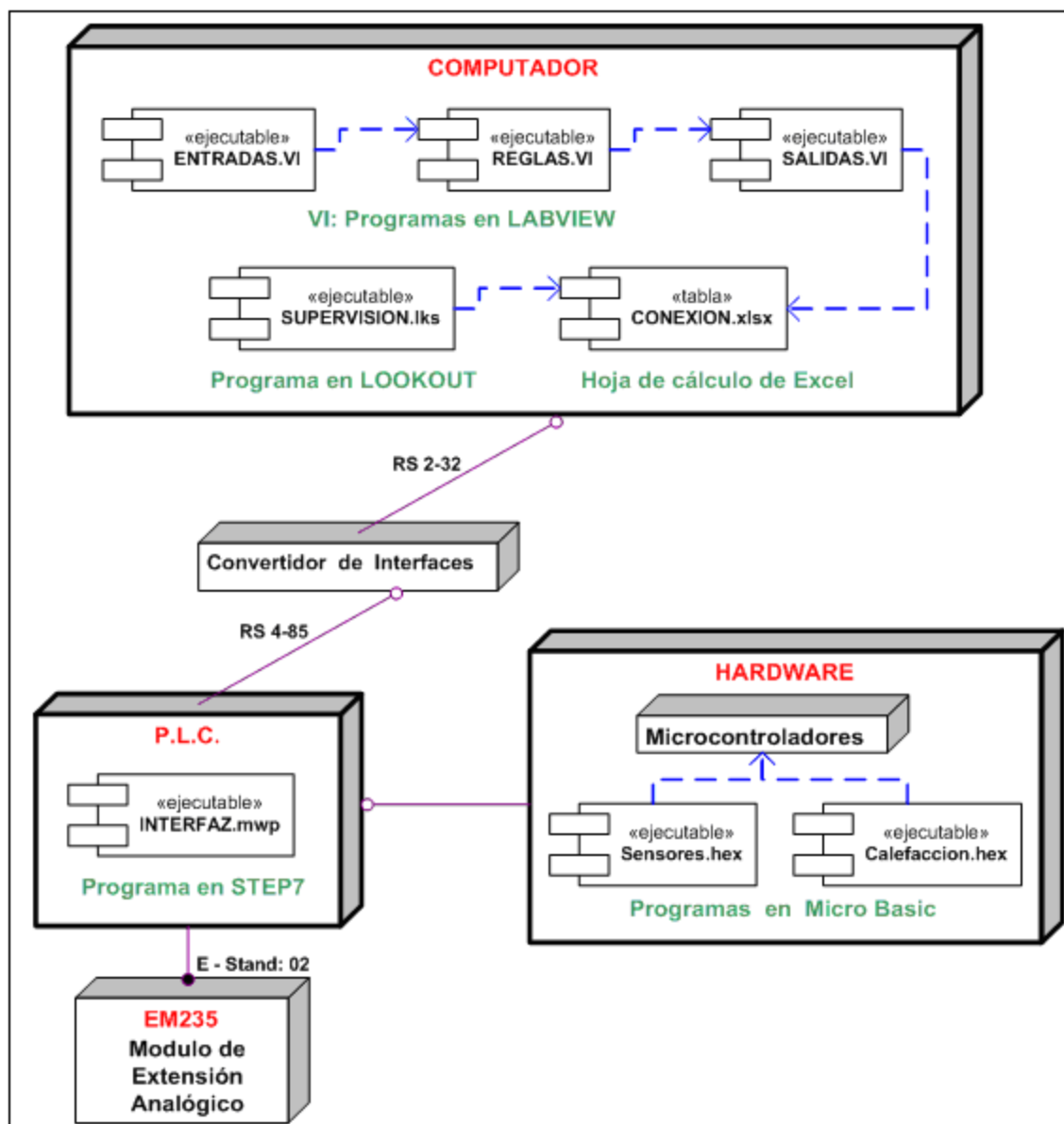


Figura 3.13 Diagrama de Componentes y Distribución 1.

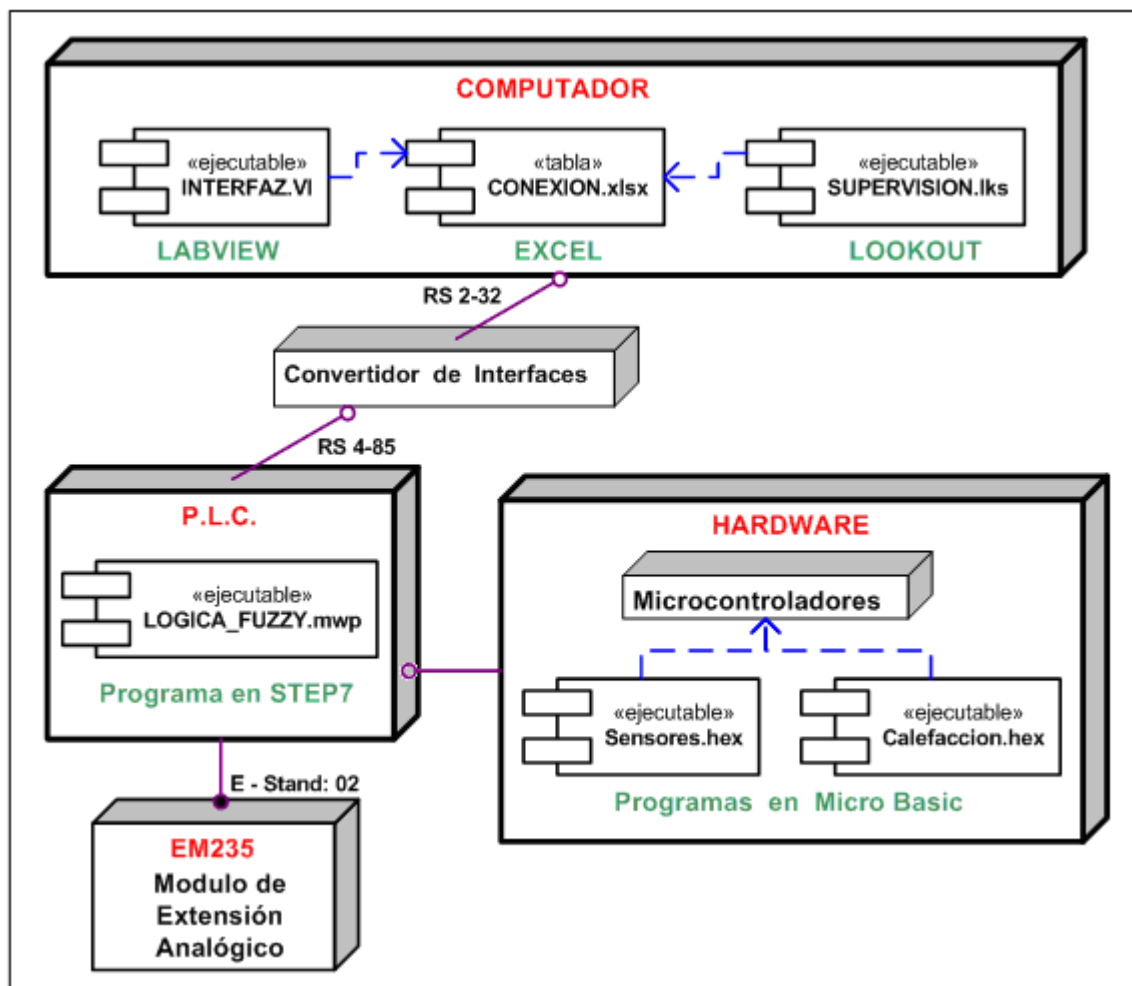


Figura 3.14 Diagrama de Componentes y Distribución 2.

3.3.2. DIAGRAMA DE CASO DE USOS.

En estos diagramas, el sistema se representa con un rectángulo, cada caso de uso se simboliza con un óvalo y los actores en forma personificada, la interacción entre los actores y los casos de uso se dibuja con líneas. Ejemplo:

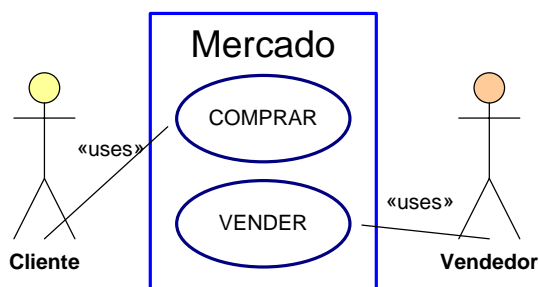


Figura 3.15 Ejemplo de diagrama de caso de usos.

A continuación el diagrama de caso de usos del proyecto:

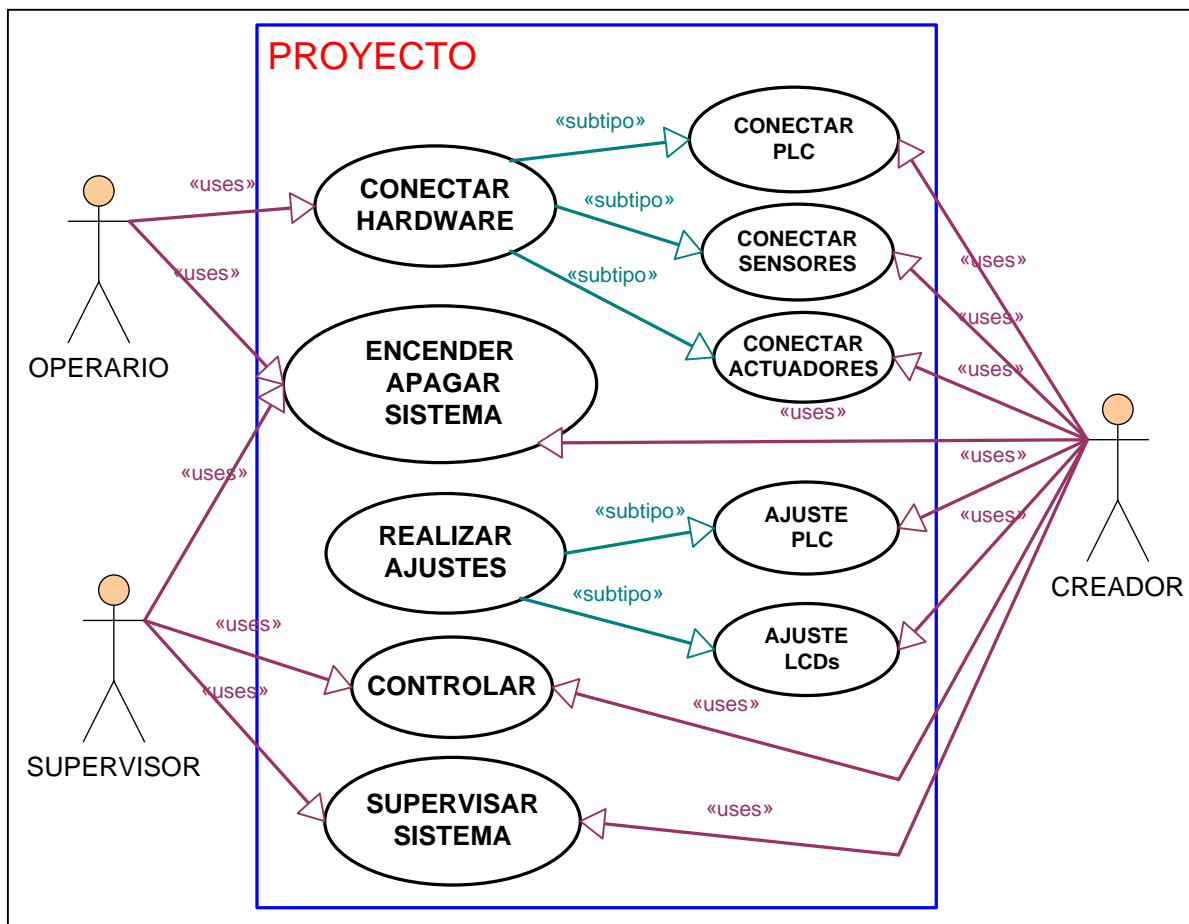


Figura 3.16 Diagrama de Caso de Usos.

3.3.3. DIAGRAMA DE ESTADOS.

Los diagramas de estado surgen por cambios en el sistema, debido a que los objetos que componen dicho sistema modificaron su *estado* como respuesta a los sucesos y al tiempo. El icono para el estado, es un rectángulo de vértices redondeados, y el símbolo de una transición es una línea continua y una punta de flecha. El círculo relleno se interpreta como el punto inicial de una secuencia de estados, y la diana representa al punto final. Se puede subdividir el estado en 3 áreas que muestren: el nombre, variables y actividades, dentro de las actividades se detallan:

- Entrada: Lo que sucede cuando el sistema entra al estado.
- Salida: Lo que ocurre cuando el sistema sale del estado.
- Hacer: Lo que pasa cuando el sistema está dentro del estado.

El proyecto consta de dos diagramas de estado, el primero para el desarrollo del proceso de lógica difusa en LabView y el segundo para el desarrollo de lógica difusa en Step7:

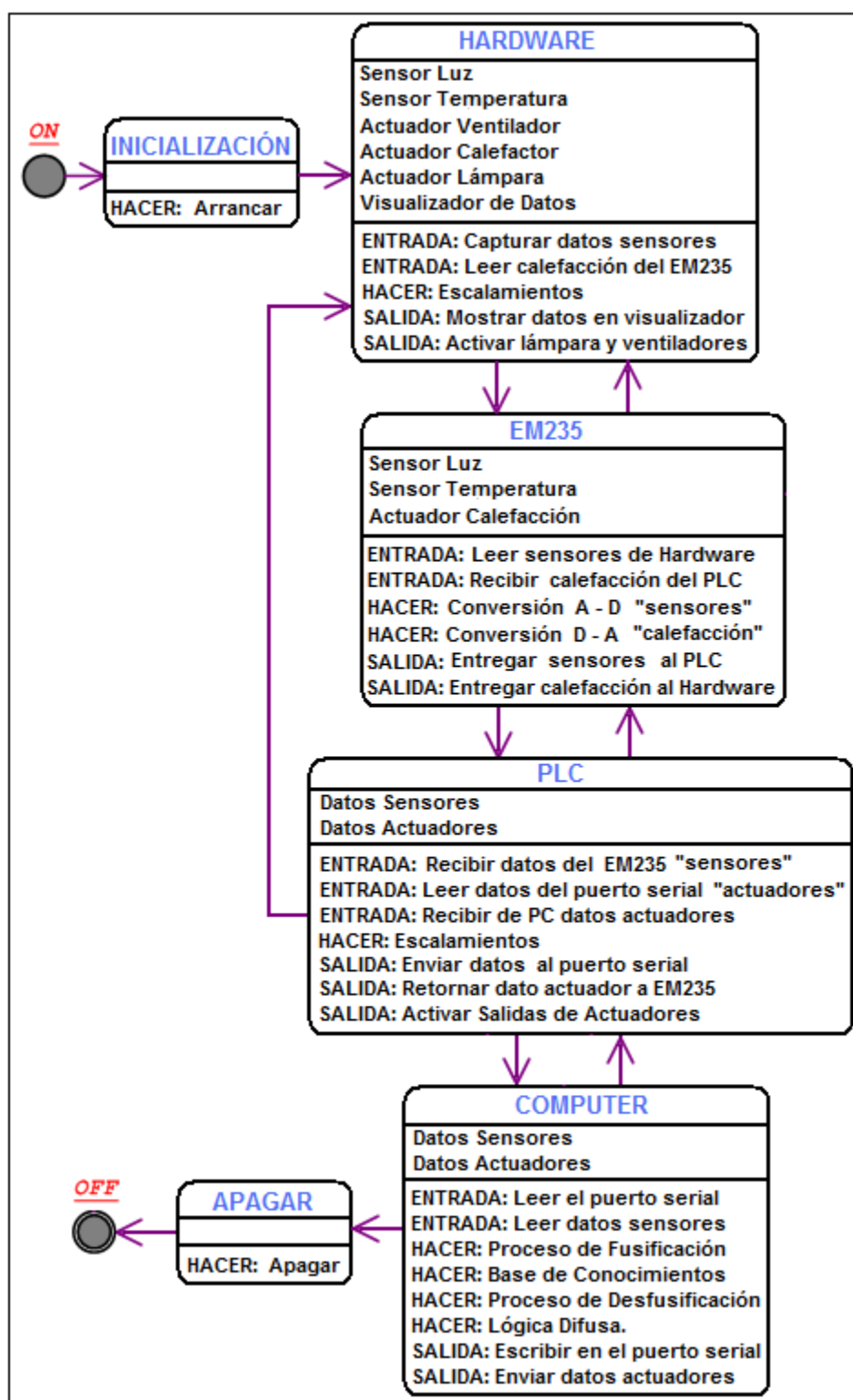


Figura 3.17 Diagrama de Estados 1.

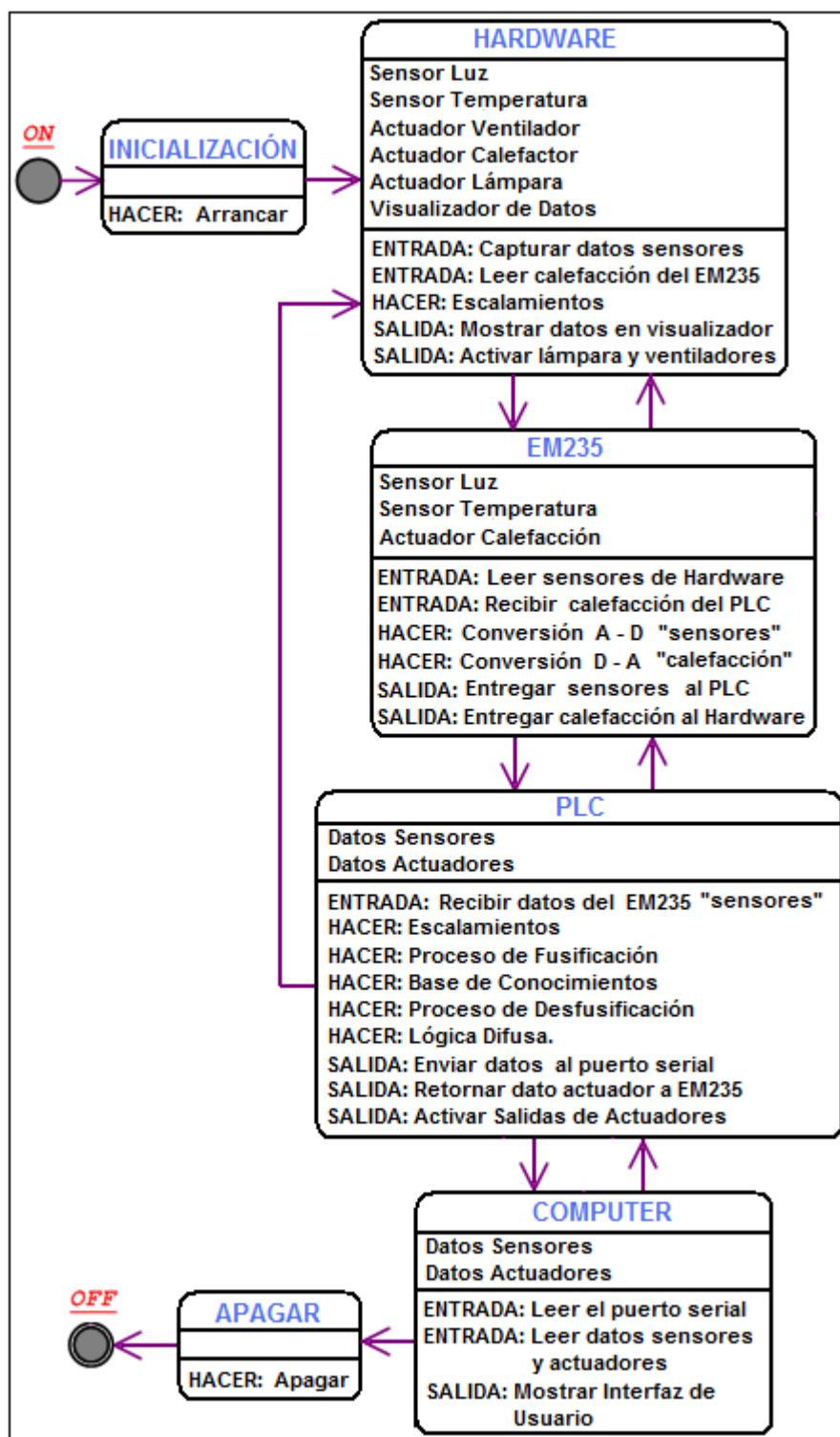


Figura 3.18 Diagrama de Estados 2.

3.3.4. DIAGRAMA DE SECUENCIA.

Este tipo de diagramas muestra una interacción ordenada según la secuencia de eventos con perspectiva a una línea de tiempo, se muestran los objetos participantes en la interacción y los mensajes ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo y en el eje horizontal se ubican los objetos y actores participantes en la interacción colocándolos de izquierda a derecha y en la parte superior.

Cada objeto o actor tiene una línea vertical y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo. Se pueden colocar etiquetas junto a las transiciones o activaciones a las que se refieren.

La línea de vida de un objeto es discontinua y se desplaza hacia abajo. Una línea continua con una punta de flecha conecta a una línea de vida con otra, representa un mensaje de un objeto a otro.

El tiempo se inicia en la parte superior y continúa hacia abajo. No necesariamente se debe especificar el tiempo de manera explícita (como en segundos, minutos, horas, días, etc.), aunque se puede hacer si resulta necesario. Siempre se tiene que ubicar correctamente en el eje vertical la secuencia correcta de eventos que se deben ir dando de forma cronológica o en una línea de tiempo.

En este caso también se debe realizar dos diagramas de secuencia correspondientes al proyecto el primero con lógica difusa en LabView y el segundo con lógica difusa en Step7:

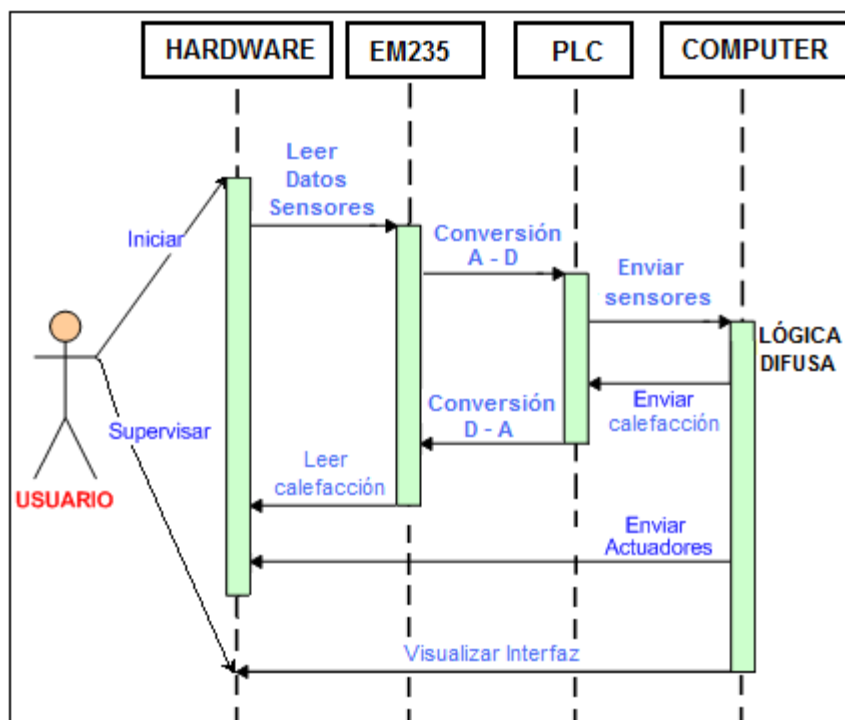


Figura 3.19 Diagrama de Secuencia 1.

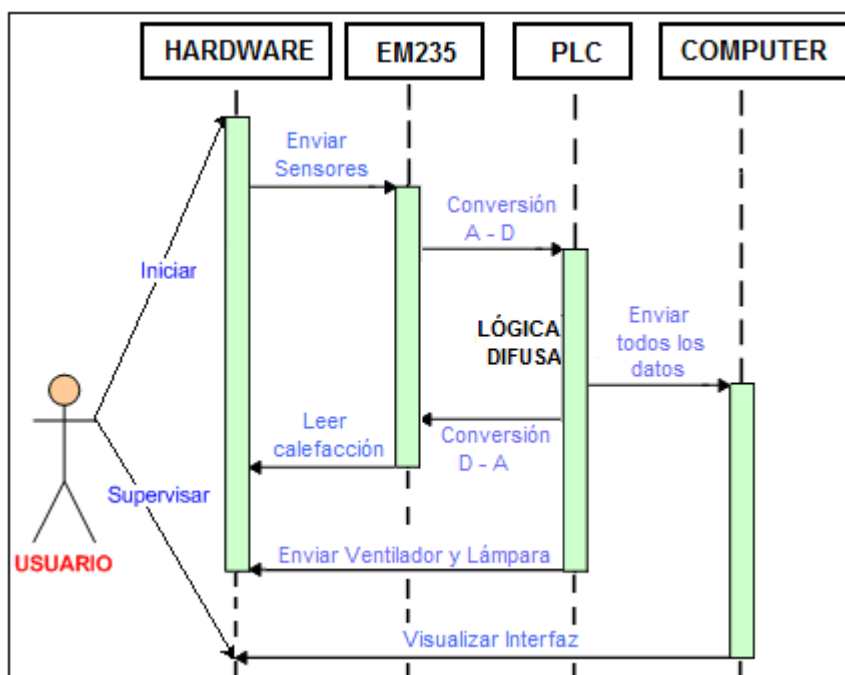


Figura 3.20 Diagrama de Secuencia 2.

3.3.5. DIAGRAMA DE BLOQUES.

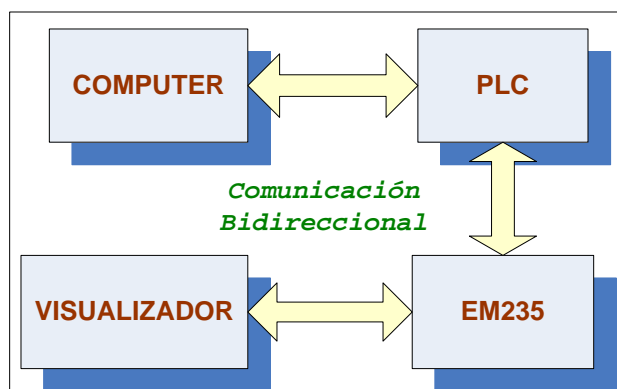


Figura 3.21 Diagrama de bloques general del proyecto.

3.3.6. DIAGRAMA DE CONECTIVIDAD.

- o Conectividad PC con convertidor de Interfaz:

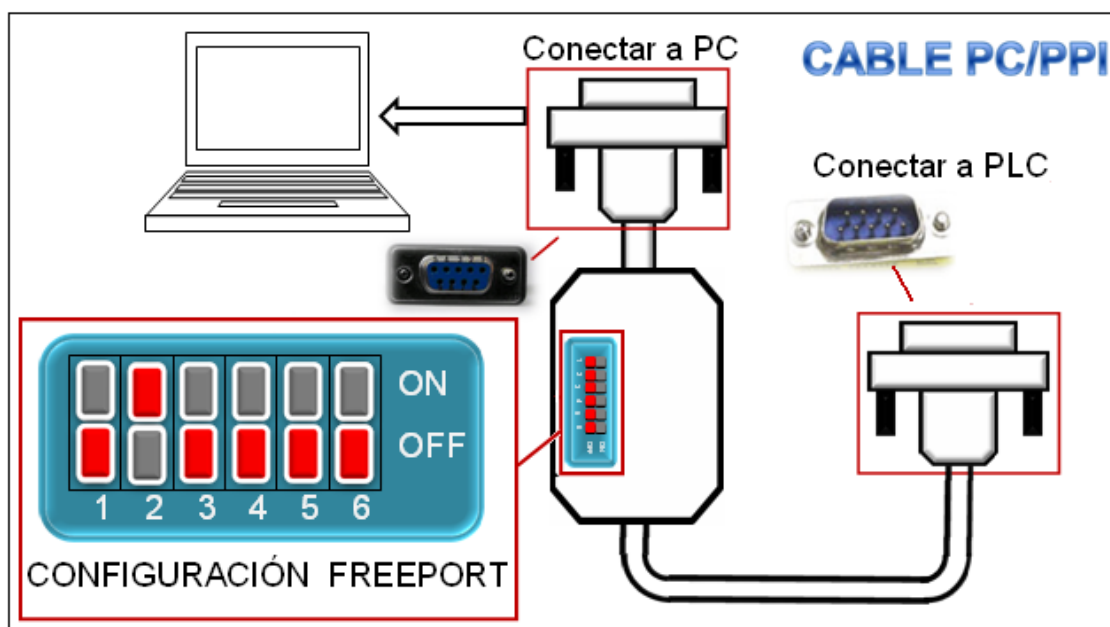


Figura 3.22 Diagrama de conexión PC cable de PLC.

- Conectividad de la conexión de alimentación eléctrica de PLC a Módulo de Extensión Analógico EM235:

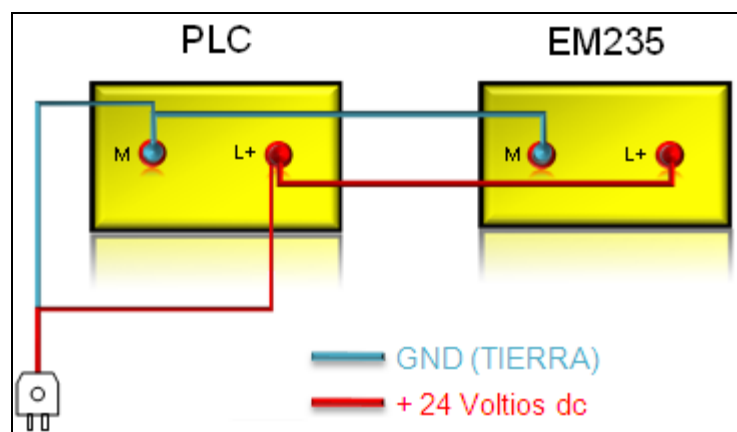


Figura 3.23 Diagrama de conexión EM235 y PLC.

- Conectividad de actuadores con PLC:

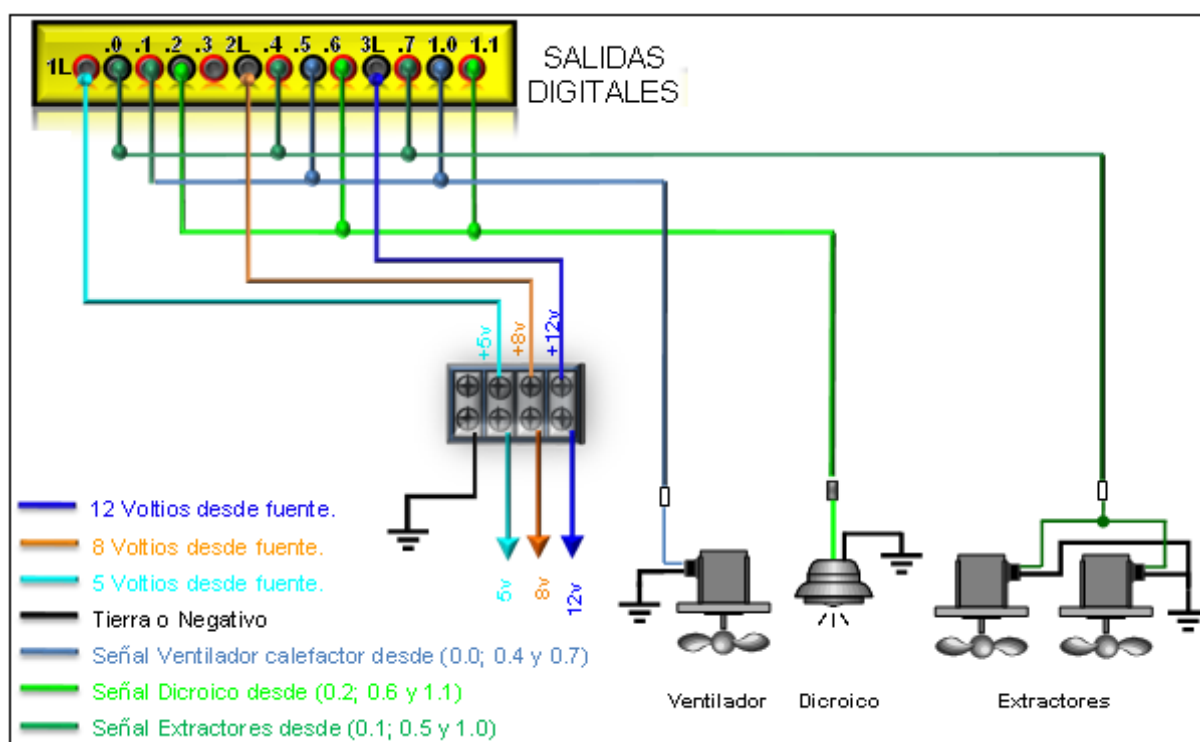


Figura 3.24 Conexión actuadores con PLC.

- Conectividad de las entradas y salida analógica con el visualizador de datos:

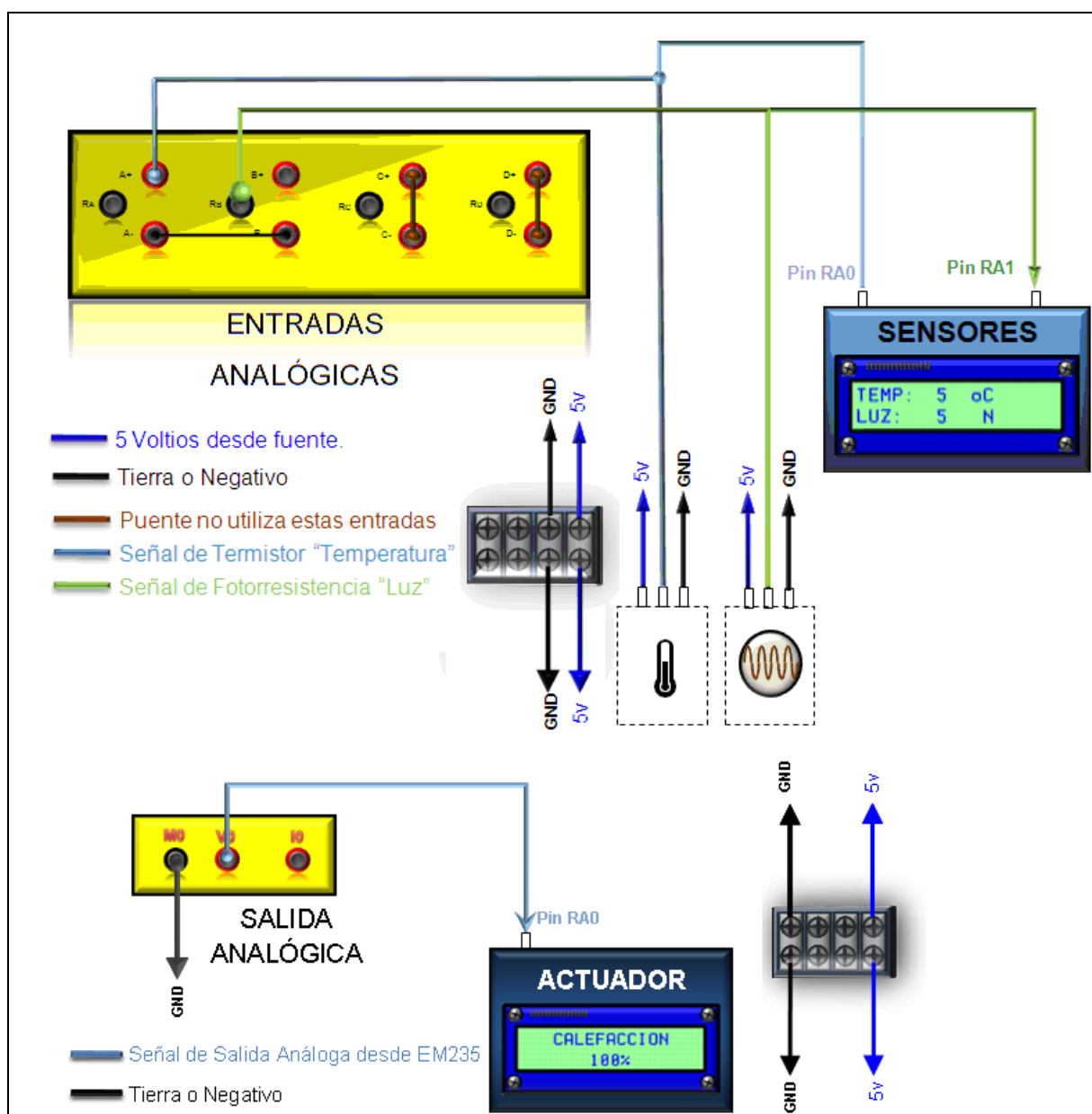


Figura 3.25 Conexión de entradas analógicas y salida con visualizador.

- Conectividad de todos los bloques:

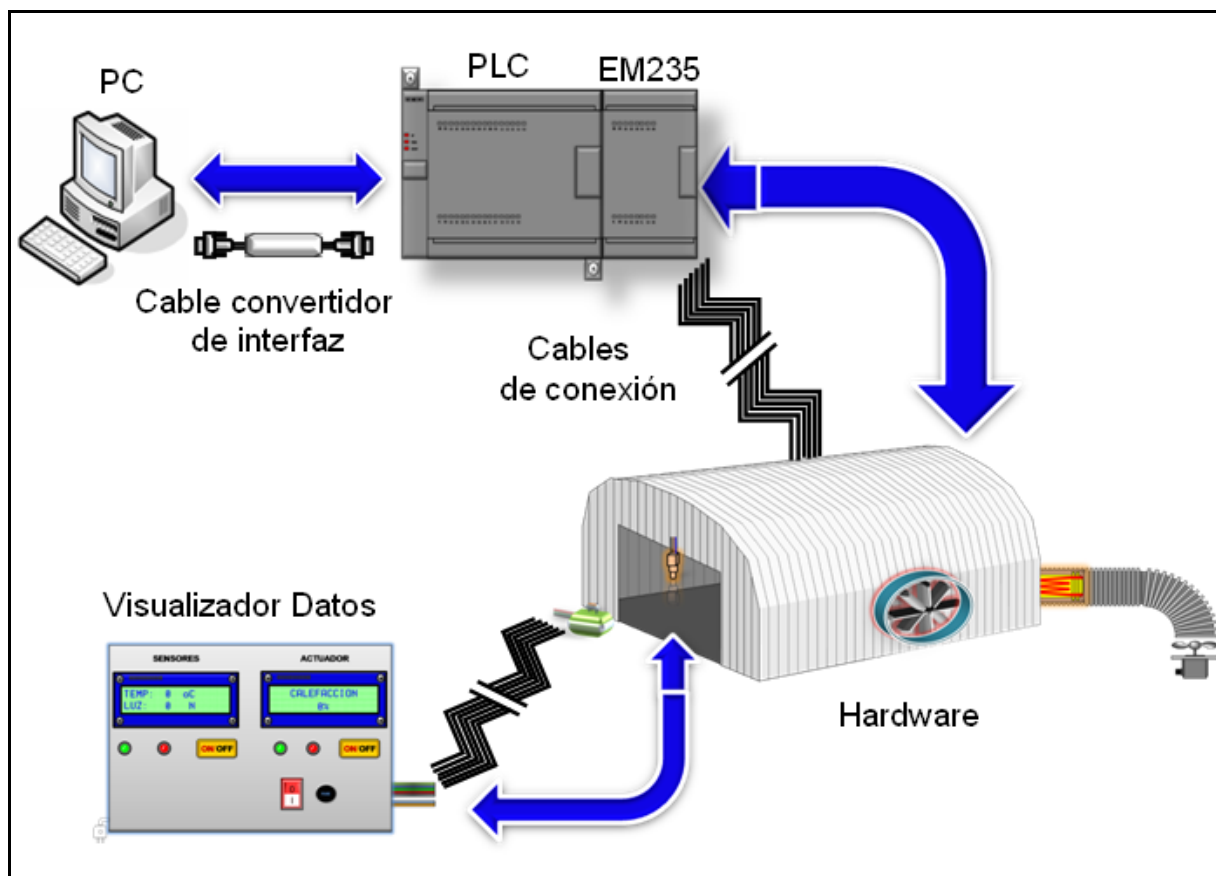


Figura 3.26 Conectividad general de todos los bloques.

3.4. PLC SIEMENS CPU 224.

La información que se presenta en este subcapítulo, es un resumen del manual de usuario de Simatic S7-200 CPU 224, este es un extracto de las características más importantes del autómata.

La familia S7 es bastante extensa, consta de: PLC S7-200, S7-300 y los S7-400. Los PLC S7-200 son accesibles en su precio, tienen funciones integradas, son expandibles hasta 7 módulos y son de tamaño pequeño. Los PLC S7-300 son escalables, se pueden conectar con una amplia gama de módulos, son expandibles hasta 32 módulos y su tamaño es medio.

Los PLC S7-400 son más rápidos su CPU trabaja a 80 nano segundos “tiempo de ejecución de cada instrucción” a diferencia de los CPU S7-200 que procesan

cada instrucción a 0.37 micro segundos; son multiprocesador, tienen alto grado de modularidad y su tamaño es grande.⁶³

La familia SIMATIC S7-200 esta compuesta de los siguientes módulos:

- a) 4 equipos básicos diferentes en diversas variantes.
- b) 14 módulos de ampliación digitales y analógicos.
- c) 2 módulos de comunicaciones para la conexión a PROFIBUS e Interfaz-AS.

También ofrece diferentes posibilidades de comunicación:

- a) La interfaz punto a punto PPI, MPI (esclavo) y en modo interfaz libremente programable como en modo Freeport.
- b) Acoplamiento a Interfaz AS, actuando como maestro a través del módulo de comunicaciones CO 243-2.
- c) Acoplamiento a PROFIBUS-DP como esclavo, a través del módulo PROFIBUS-DP EM 277.

El proyecto se realiza con un PLC S7-200 de CPU 224 porque nos brinda las características necesarias y suficientes como por ejemplo capacidad de conexión con un módulo analógico y protocolo de comunicación Freeport.

Este micro-PLC cuenta con una alta escala de integración, aplicable tanto para los controles más simples como también para tareas complejas de automatización. Puede implementarse aislado, interconectado en red o en configuraciones descentralizadas.

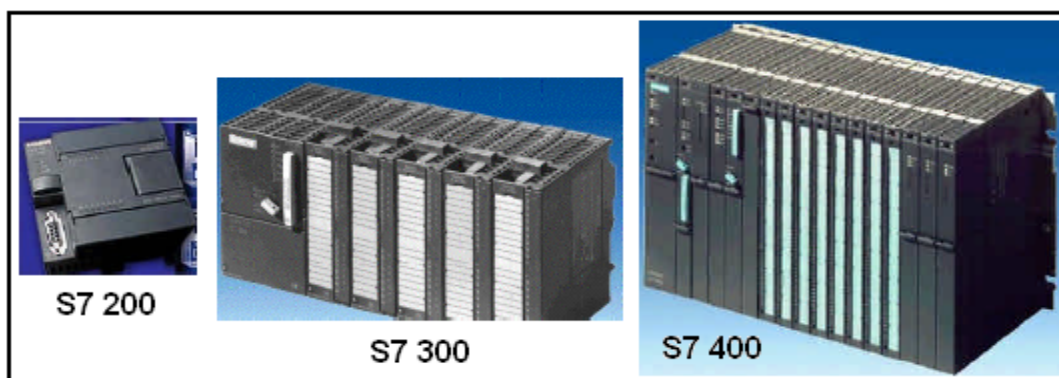


Figura 3.27 Modelos de PLC Siemens.

⁶³ http://isa.umh.es/asignaturas/asct/practicas_automatas/practical1_siemens.pdf Manual de usuario de PLC Simatic S7 200.

3.4.1. CARACTERÍSTICAS.

El PLC S7-200 con CPU 224 es de alto nivel, está compuesto de 14 entradas digitales y 10 salidas digitales tipo relé, las entradas y salidas son ampliables, su memoria para almacenar el programa de usuario es de 8 Kbytes y la memoria para los datos de usuario es de 5 Kbytes en versión 1.22 y 8 Kbytes en versión 2.0, sus dimensiones son: 120 x 80 x 62 milímetros. Además es modular, se pueden conectar hasta 7 módulos. Ahorran espacio por su tamaño y porque se pueden conectar en dos filas.

La imagen de proceso de entradas es desde I0.0 hasta I15.7 y de las salidas es desde Q0.0 hasta Q15.7, soporta módulos analógicos trabajando con 63 entradas y 63 salidas analógicas: AIW0 – AIW62 y AQW0 – AQW62. Tiene una memoria de variables de: VB0 a VB5119 en versión 1.22 y VB0 a VB8119 en versión 2.0.

Tiene una memoria local de LB0 a LB59, el área de marcas es desde M0.0 a M31.7 las marcas especiales van desde SM0.0 hasta SM549.7

Contiene 256 temporizadores y 256 contadores: T0 - T255 y C0 – C255, tiene 4 acumuladores: AC0 – AC3. Se pueden crear hasta 64 subrutinas y 128 interrupciones; se pueden utilizar 256 flancos, 8 lazos PID.

Se comunica con la PC a través del puerto serial de comunicación COM 0.

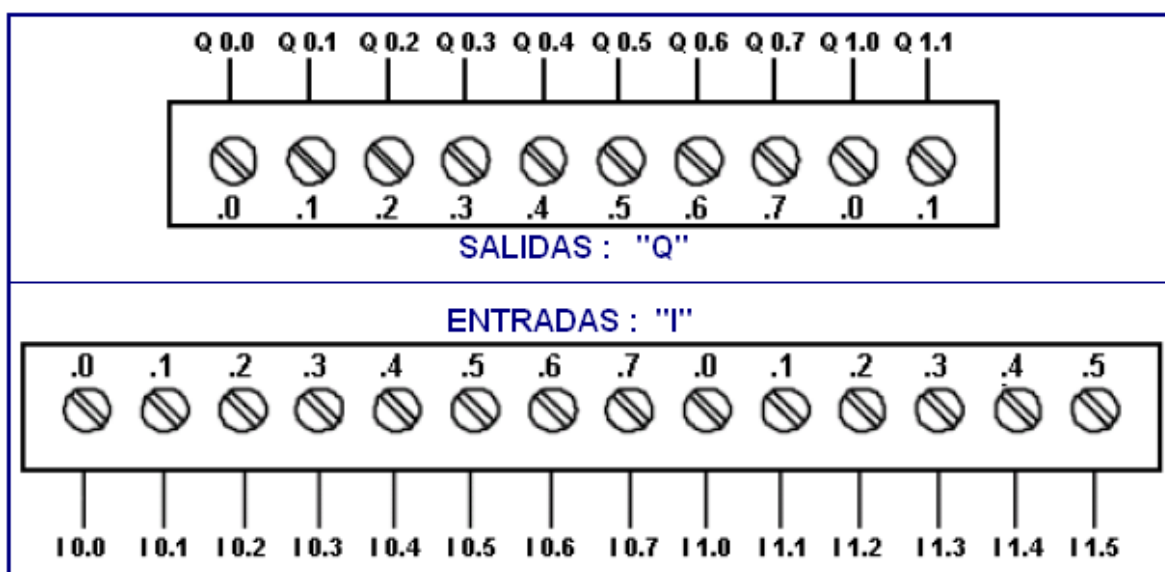


Figura 3.28 Entradas y Salidas del CPU 224.

3.4.2. COMPONENTES.

1. Salidas digitales integradas.
2. LEDs de estado de las salidas digitales.
3. Terminales de alimentación.
4. Conmutador Stop/Run.
5. Conector para el cable de ampliación.
6. LEDs de estado de la CPU.
7. Ranura para el cartucho de memoria.
8. Puerto de comunicaciones (p. Ej. PPI).
9. Entradas digitales integradas.
10. LEDs de estado de las entradas digitales.
11. Fuente de alimentación integrada.
12. Potenciómetros integrados.
13. Módulo de ampliación.
14. Fijadores para tornillo (DIN métrica M4, diámetro 5 mm).
15. Pestaña de fijación.

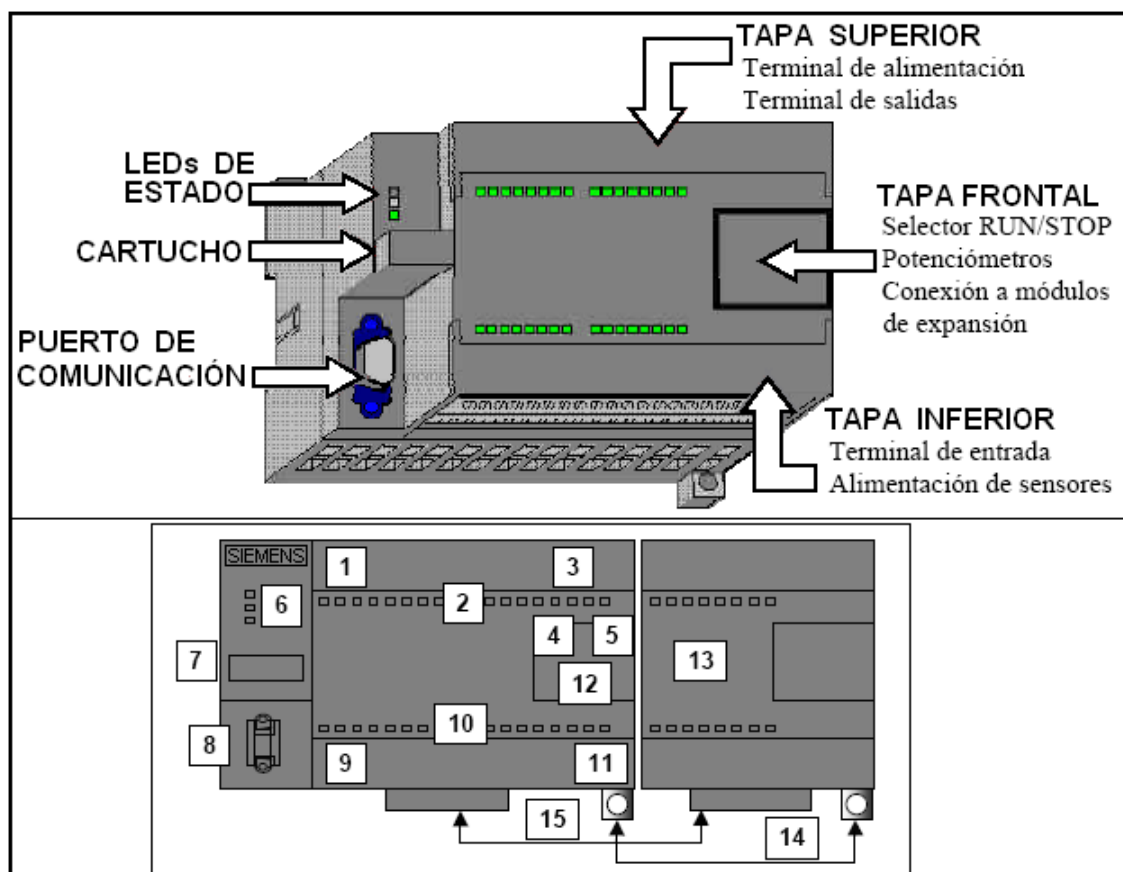


Figura 3.29 Componentes del CPU 224.

El montaje del PLC se puede realizar sobre carril estándar por ejemplo carril DIN, o montaje directo sobre pared a través de taladros. El CPU 224 ahorra espacio en el montaje gracias al pequeño tamaño de los componentes.

3.4.3. CABLE PC/PPI

El cable PC/PPI se utiliza para la conexión del PLC con la computadora, impresoras, módems, etc. El cable es una interfaz que transforma: RS 485 (PLC) a RS 232 (PC).

La velocidad admitida: 1,2 KBaud - 38,4 KBaud.

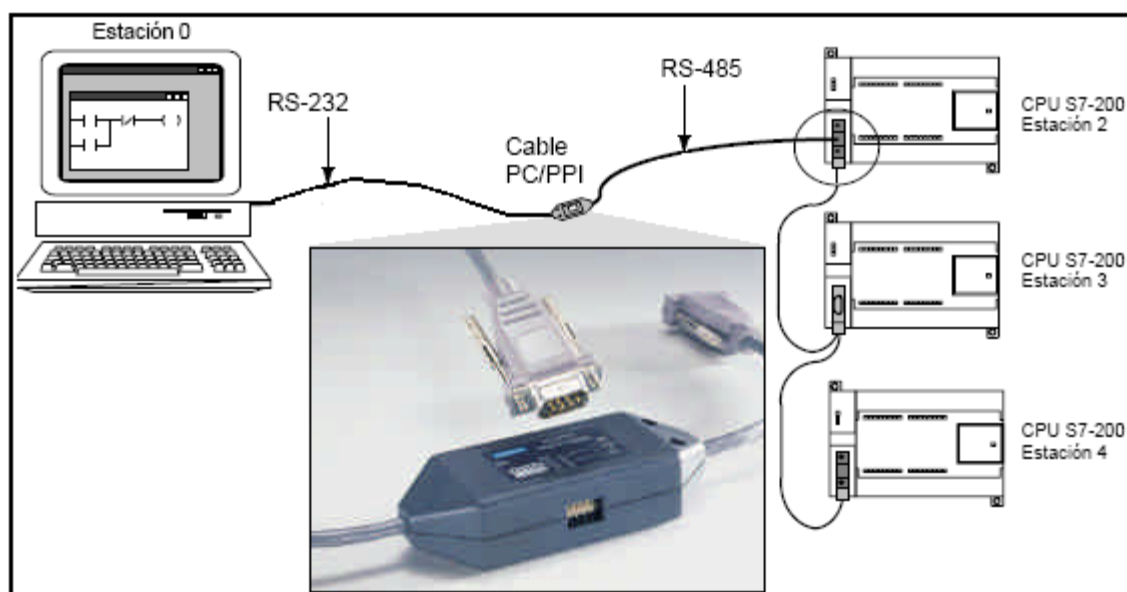


Figura 3.30 Interfaz PC/PPI.

El nuevo cable PC/PPI, incorpora potencial separado mediante aislamiento óptico, permite establecer comunicación entre las CPUs S7-22X y módems de 10 bits. Contiene un switch para conmutar entre los modos 10/11 bits, además contiene un adaptador de módem nulo integrado.

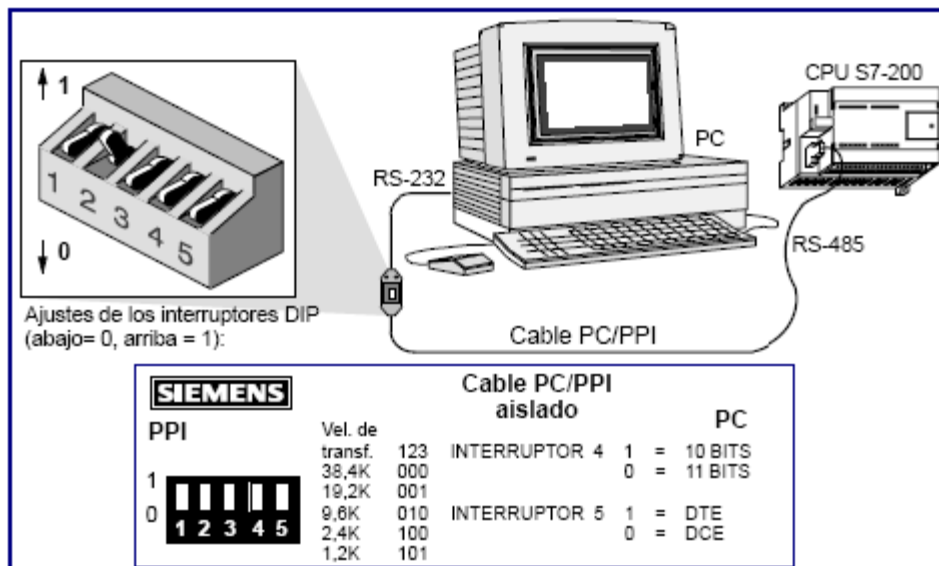


Figura 3.31 Configuración del DIP en el cable PC/PPI.⁶⁴

3.4.4. CONCEPTOS BÁSICOS.

- **Bit:** Unidad del símbolo binario, solamente puede tomar los valores decimales: "0" y "1". En ocasiones, el bit es insuficiente para definir determinados aspectos de una automatización. Debiendo recurrir a conjuntos formados por varios símbolos binarios.
- **Byte:** Conjunto de 8 símbolos binarios, el byte tiene una longitud de 8 bits, cada uno de lo cuales puede tomar cualquier valor entre 0 y 1. En decimal un byte puede tomar diferentes valores, pero sólo hasta 255.
- **Palabra:** En un PLC los bits se asocian en grupos y cada bit en dicho grupo está exactamente definido por una posición propia que tiene una dirección específica. Un byte tiene una dirección de byte y direcciones de bit 0...7. El formato palabra está compuesto por 2 bytes ó 16 bits. En un PLC una palabra permite representar valores numéricos de -32767 a +32767, ó 65535 sólo números positivos.

Se ha convenido que el bit con el peso 2^{15} señala números negativos (si aparece un "1" en la posición 2^{15} , el número en cuestión es negativo).

⁶⁴ <http://www.salesianos.edu/alcoy.juanxxiii/dpts/docs/automatas.pdf> Configuración del interruptor DIP en el cable PC/PPI.

Direccionamiento: El autómatas utiliza un operando distintivo:

I para denominar entradas (algunos lenguajes utilizan la *E*), **Q** para denominar salidas (algunos lenguajes utilizan la *A*). Junto con el distintivo de entrada o salida aparece el parámetro 0.4, 1.2 ó 4.7. El parámetro consiste en una combinación:

0, 1, 4 → byte. 4, 2, 7 → bit.

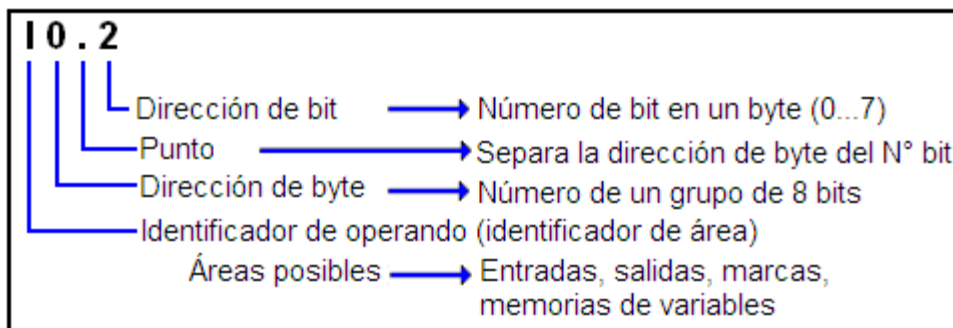


Figura 3.32 Estructura de un operando.

En caso de marcas se llama con la letra M, en marcas espaciales con SM, en variables la letra V seguida de la letra de memoria, por ejemplo en formato byte con: VB, en palabra: VW, en doble palabra: VD.

El formato doble palabra consta de 4 bytes ó 32 bits. Sirve para trabajar con decimales o números reales o con enteros de 32 bits, a diferencia de la palabra que trabaja con enteros de 16 bits.

Ejemplos: VB0, VB1, VB2, VB2000, VW0, VW2, VW4, VW6, VW1000, VD4, VD8, VD12, VD16, VD500, etc. Los bytes se llaman de uno en uno, las palabras se llaman de dos en dos, las palabras dobles se llaman de cuatro en cuatro; esto se hace para que no exista solapamiento en la memoria.

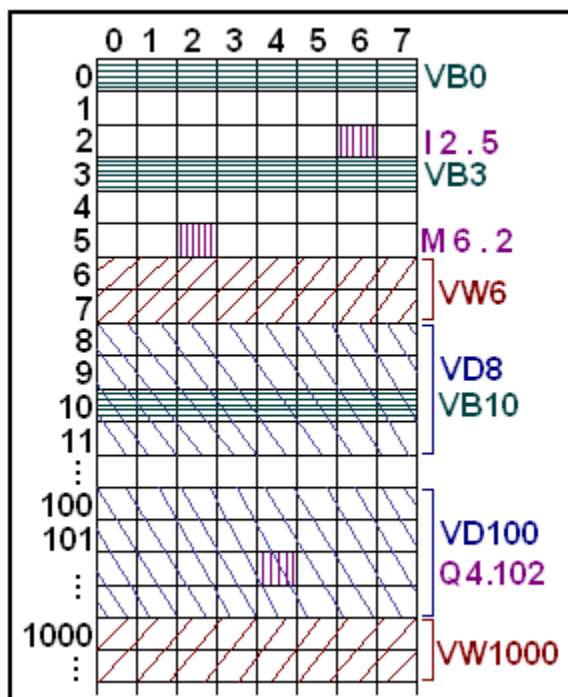


Figura 3.33 Asignación de memoria.

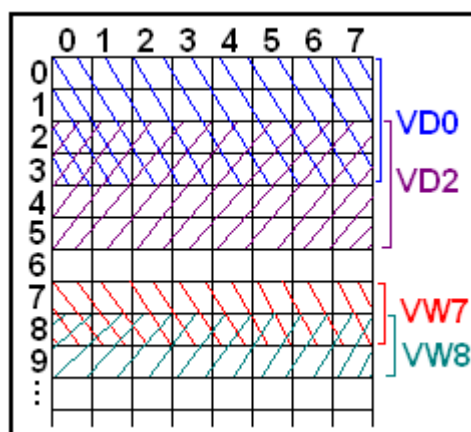


Figura 3.34 Solapamiento de variables.

3.4.5. CONEXIÓN DE ENTRADAS Y SALIDAS.

En el PLC hay bornera para las entradas, en donde se conectarán todos los switches, interruptores, pulsadores, sensores, etc y bornera para las salidas, en donde se conectarán los actuadores como ventiladores, calefactores, lámparas, válvulas, motores, etc.

En el presente proyecto como entradas se conectan los sensores de luminosidad y temperatura de aire y como actuadores se conectan dos ventiladores, un calefactor y una lámpara.

Las conexiones para las entradas se muestran en el siguiente gráfico:

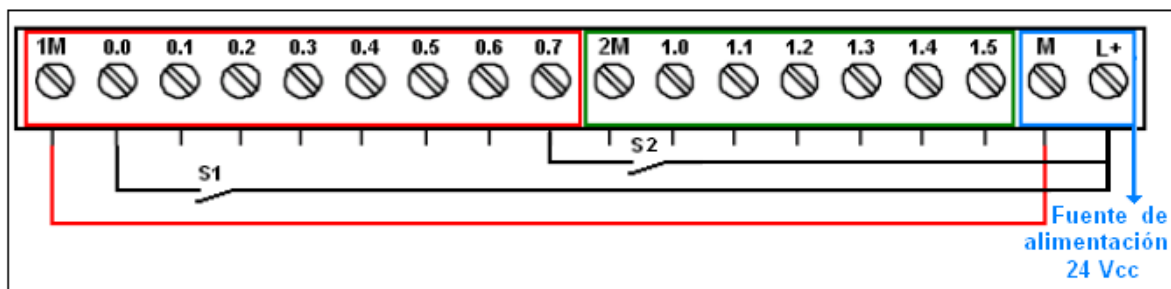


Figura 3.35 Conexión de las entradas.

Se debe realizar un puente entre 1M y M para alimentar el conjunto, es una especie de permiso de conexión, para activar el conjunto 2M se debe realizar un puente entre 2M y M. En el caso de que el sensor necesitara alimentación externa, se debe conectar la fuente a 1M o a 2M siempre que el bloque contenga sensores que trabajen con el mismo voltaje.

Para las salidas la conexión se muestra en la siguiente figura:

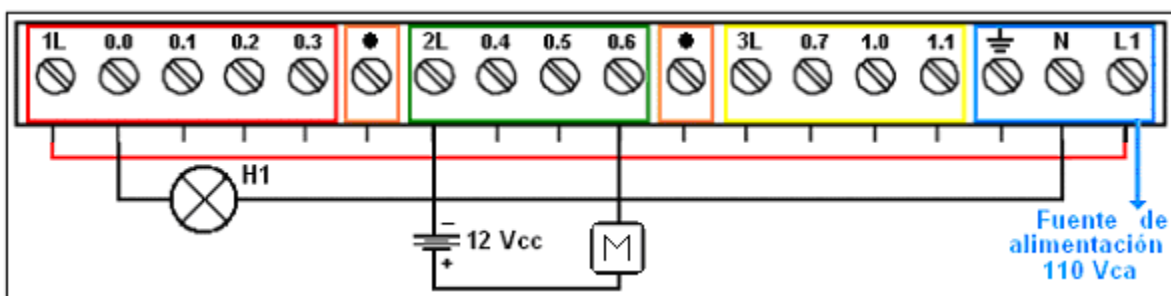


Figura 3.36 Conexión de las salidas.

Para activar cada conjunto, se debe conectar el positivo de alimentación L1 para 110 V de corriente alterna a 1L, 2L ó 3L según convenga.

Al realizar este conexionado, se tiene todas las salidas asociadas a 1L desde 0.0 hasta 0.3 alimentadas a 110 V en alterna, por tanto hay que tener cuidado de no conectar a estas salidas ningún dispositivo que funcione a otra tensión, por lo que se debe utilizar en este caso una salida de otro conjunto por ejemplo 0.4 perteneciente a 2L, alimentando dicho conjunto a la tensión apropiada. Las salidas identificadas con un punto no tienen conexión es decir están deshabilitadas.

3.4.6. DATOS TÉCNICOS DE LA CPU 224.

Algunos datos técnicos del PLC que se utiliza en el proyecto, se muestran a continuación. Los datos sombreados corresponden a la CPU 224 utilizada con el número: 214-1BD22-0XB0.

Nº de referencia	Modelo de CPU	Alimentación (nominal)	Entradas de la CPU	Salidas de la CPU	terminal extraíble
6ES7 211-0AA22-0XB0	CPU 221	DC 24 V	6 x DC 24 V	4 x DC 24 V	No
6ES7 211-0BA22-0XB0	CPU 221	AC 120 a 240 V	6 x DC 24 V	4 salidas de relé	No
6ES7 212-1AB22-0XB0	CPU 222	DC 24 V	8 x DC 24 V	6 x DC 24 V	No
6ES7 212-1BB22-0XB0	CPU 222	AC 120 a 240 V	8 x DC 24 V	6 salidas de relé	No
6ES7 214-1AD22-0XB0	CPU 224	DC 24 V	14 x DC 24 V	10 x DC 24 V	Sí
6ES7 214-1BD22-0XB0	CPU 224	AC 120 a 240 V	14 x DC 24 V	10 salidas de relé	Sí
6ES7 216-2AD22-0XB0	CPU 226	DC 24 V	24 x DC 24 V	16 x DC 24 V	Sí
6ES7 216-2BD22-0XB0	CPU 226	AC 120 a 240 V	24 x DC 24 V	16 salidas de relé	Sí
6ES7 216-2AF22-0XB0	CPU 226XM	DC 24 V	24 x DC 24 V	16 x DC 24 V	Sí
6ES7 216-2BF22-0XB0	CPU 226XM	AC 120 a 240 V	24 x DC 24 V	16 salidas de relé	Sí

Figura 3.37 Números de referencia de las CPUs.

Nº de referencia	Nombre y descripción de la CPU	Dimensiones en mm (l x a x p)	Peso	Disipación	Tensión DC disponible	
					DC +5 V	DC +24 V
6ES7 211-0AA22-0XB0	CPU 221 DC/DC/DC 6 entradas/4 salidas	90 x 80 x 62	270 g	3 W	0 mA	180 mA
6ES7 211-0BA22-0XB0	CPU 221 AC/DC/relé 6 entradas/4 salidas de relé	90 x 80 x 62	310 g	6 W	0 mA	180 mA
6ES7 212-1AB22-0XB0	CPU 222 DC/DC/DC 8 entradas/6 salidas	90 x 80 x 62	270 g	5 W	340 mA	180 mA
6ES7 212-1BB22-0XB0	CPU 222 AC/DC/relé 8 entradas/6 salidas de relé	90 x 80 x 62	310 g	7 W	340 mA	180 mA
6ES7 214-1AD22-0XB0	CPU 224 DC/DC/DC 14 entradas/10 salidas	120,5 x 80 x 62	360 g	7 W	660 mA	280 mA
6ES7 214-1BD22-0XB0	CPU 224 AC/DC/relé 14 entradas/10 salidas de relé	120,5 x 80 x 62	410 g	10 W	660 mA	280 mA
6ES7 216-2AD22-0XB0	CPU 226 DC/DC/DC 24 entradas/16 salidas	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BD22-0XB0	CPU 226 AC/DC/relé 24 entradas/16 salidas de relé	196 x 80 x 62	660 g	17 W	1000 mA	400 mA
6ES7 216-2AF22-0XB0	CPU 226XM DC/DC/DC 24 entradas/16 salidas	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BF22-0XB0	CPU 226XM AC/DC/relé 24 entradas/16 salidas de relé	196 x 80 x 62	660 g	17 W	1000 mA	400 mA

Figura 3.38 Datos de las características generales de las CPUs.

Memoria	
Tamaño del programa de usuario (EEPROM)	4096 palabras
Datos de usuario (EEPROM)	2560 palabras (remanentes)
Respaldo (condensador de alto rendimiento) (pila opcional)	Típ. 190 h (mín. 120 h a 40° C) Típ. 200 días
Entradas y salidas (E/S)	
E/S digitales incorporadas	14 E/10 S
Tamaño de la imagen de E/S analógicas	64 (32 E/32 S)
Nº máx. de módulos de ampliación	7 módulos
Nº máx. de módulos de ampliación inteligentes	7 módulos
Entradas de captura de impulsos	14
Contadores rápidos Fase simple 2 fases	6 contadores en total 6 a 30 kHz 4 a 20 kHz
Potenciómetros analógicos	2 con resolución de 8 bits
Reloj de tiempo real	Incorporado
Cartuchos opcionales	Memoria y pila

Figura 3.39 Características de la CPU 224.

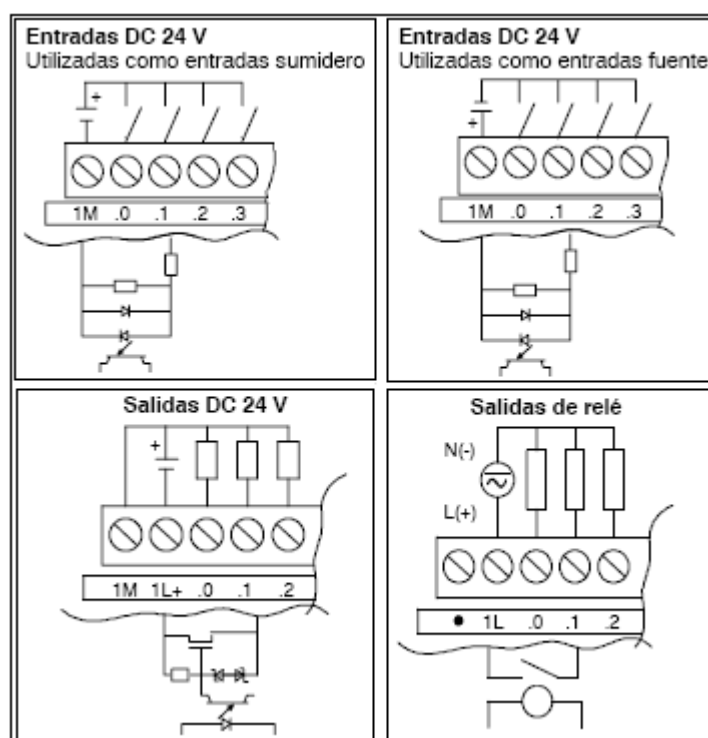


Figura 3.40 Cableado para la conexión de entradas y salidas.

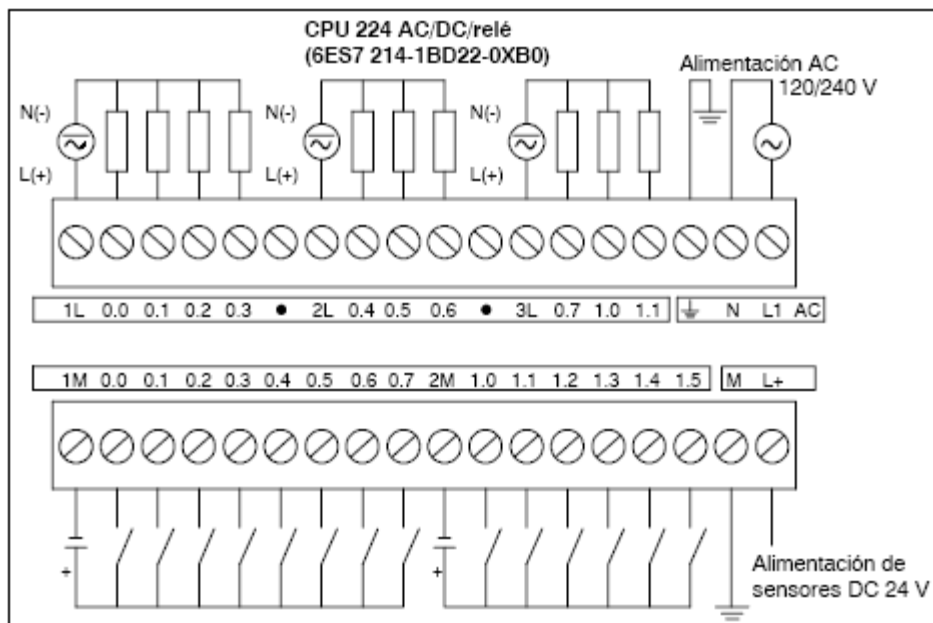


Figura 3.41 Diagrama de cableado de la CPU 224.

3.5. MÓDULO ANALÓGICO EM-235.

El módulo de expansión analógico EM-235 tiene 1 salida analógica y 4 entradas analógicas; que se pueden configurar para funcionar en los rangos -10V a +10V en modo bipolar o de 0 a 10V en modo unipolar.

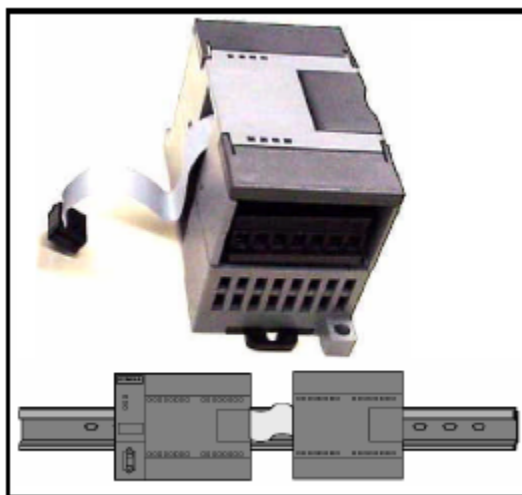


Figura 3. 42 Módulo EM-235.

Las entradas analógicas del módulo transforman señales analógicas del proceso en señales binarias para su tratamiento dentro del SIMATIC S7-200. Las salidas

analógicas del módulo transforman señales binarias del S7-200 en señales analógicas para el proceso.⁶⁵

A las entradas analógicas se puede conectar sensores y emisores de señal de tipo tensión o intensidad, resistencia, así como termopares y termo resistencias.

Es una herramienta indispensable para convertidores de frecuencias, regulaciones, etc además dispone de 2, 4 u 8 canales y tiene una resolución de 4 y 8 bits, con posibilidad de configuración para señales tipo tensión o corriente.

Para acceder a las entradas/salidas analógicas, se pueden utilizar las borneras presentes en el entrenador o bien utilizar directamente las regletas de conexiones de los propios autómatas “hileras inferior y superior”. La forma de acceder a cada entrada/salida analógicas se muestra en la tabla siguiente:

Señal	Conectores en el entrenador (+ / -)	Conectores en regleta (autómata)
Entrada 0	AIW0+ / AIW0-	A+ / A-
Entrada 1	AIW2+ / AIW2-	B+ / B-
Entrada 2	AIW4+ / AIW4-	C+ / C-
Entrada 3	AIW6+ / AIW6-	D+ / D-
Salida	AQW0+ / AQW0-	V0 / M0

Figura 3.43 Acceder a E/S del EM-235.⁶⁶

El montaje del módulo es prácticamente igual que el autómata, se puede montar en el riel DIN, o en dos filas mediante cable de conexión flexible.

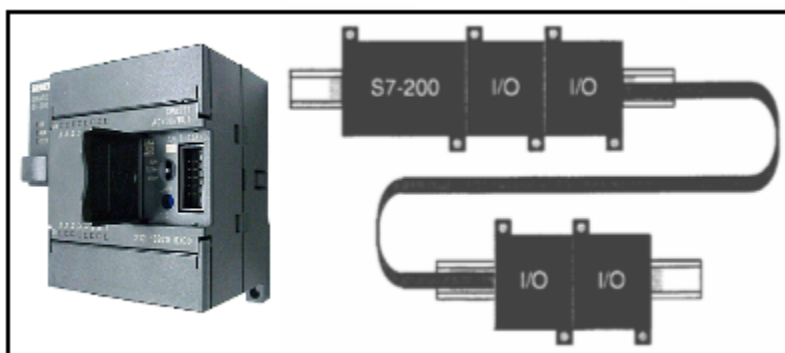


Figura 3.44 Montaje del módulo de expansión.

⁶⁵ http://isa.umh.es/asignaturas/asct/practicas_automatas/practical_siemens.pdf Módulo de expansión analógico EM235.

⁶⁶ <http://isa.umh.es/asignaturas/ai/practicas/p10.pdf> Acceder a entradas y salidas del módulo de expansión EM-235.

Es necesario comprobar que el módulo EM-235 se configure adecuadamente: los switches (DIP) accesibles en la parte inferior del módulo deben tener la combinación de valores que se necesite.



Figura 3.45 Configuración de entradas y salidas analógicas de 0 a 5 voltios.

3.5.1. CARACTERÍSTICAS DEL MÓDULO ANALÓGICO EM-235.

Las características del módulo de extensión se detallan en las siguientes figuras, el módulo que se utiliza corresponde al serial: 6ES7 235-0KD22-0XA0.

Nº de referencia	Nombre y descripción	Dimensiones en mm (l x a x p)	Peso	Disipación	Tensión DC disponible	
					DC +5 V	DC +24 V
6ES7 231-0HC22-0XA0	EM 231, 4 entradas analógicas	71,2 x 80 x 62	183 g	2 W	20 mA	60 mA
6ES7 232-0HB22-0XA0	EM 232, 2 salidas analógicas	46 x 80 x 62	148 g	2 W	20 mA	70 mA (ambas salidas a 20 mA)
6ES7 235-0KD22-0XA0	EM 235, 4 entradas analógicas/1 salida analógica	71,2 x 80 x 62	186 g	2 W	30 mA	60 mA (salida a 20 mA)

Figura 3.46 Características generales de los módulos analógicos.

Datos generales		6ES7 235-0KD22-0XA0
Aislamiento (campo a circuito lógico)		Ninguno
Rango de señales		
Salida de tensión		± 10 V
Salida de intensidad		0 a 20 mA
Resolución, rango máx.		
Tensión		12 bits
Intensidad		11 bits
Formato de la palabra de datos		
Tensión		-32000 a +32000
Intensidad		0 a +32000
Precisión		
Caso más desfavorable, 0° a 55° C		
Salida de tensión		$\pm 2\%$ de rango máx.
Salida de intensidad		$\pm 2\%$ de rango máx.
Típico, 25° C		
Salida de tensión		$\pm 0,5\%$ de rango máx.
Salida de intensidad		$\pm 0,5\%$ de rango máx.
Tiempo de ajuste		
Salida de tensión		100 μ S
Salida de intensidad		2 mS
Accionamiento máx.		
Salida de tensión		Mín. 5000 Ω
Salida de intensidad		Máx. 500 Ω

Figura 3.47 Datos de la salida del EM235

Datos generales	6ES7 235-0KD22-0XA0
Formato de la palabra de datos Bipolar, rango máx. Unipolar, rango máx.	-32000 a +32000 0 a 32000
Impedancia de entrada DC	$\geq 10\text{ M}\Omega$ entrada de tensión, 250 Ω entrada de intensidad
Atenuación filtro de entrada	-3 db a 3,1 kHz
Tensión de entrada máx.	DC 30 V
Intensidad de entrada máx.	32 mA
Resolución	Convertidor A/D de 12 bits
Aislamiento (campo a circuito lógico)	Ninguno
Tipo de entrada	Diferencial
Rangos de las entradas Tensión (unipolar)	0 a 10 V, 0 a 5 V, 0 a 1 V, 0 a 500 mV, 0 a 100 mV, 0 a 50 mV
Tensión (bipolar)	$\pm 10\text{ V}$, $\pm 5\text{ V}$, $\pm 2.5\text{ V}$, $\pm 1\text{ V}$, $\pm 500\text{ mV}$, $\pm 250\text{ mV}$, $\pm 100\text{ mV}$, $\pm 50\text{ mV}$, $\pm 25\text{ mV}$
Intensidad	0 a 20 mA
Tiempo de conversión analógica/digital	< 250 μs
Respuesta de salto de la entrada analógica	1,5 ms a 95%
Rechazo en modo común	40 dB, DC a 60 Hz
Tensión en modo común	Tensión de señal más tensión en modo común (debe ser $\leq \pm 12\text{ V}$)
Rango de tensión de alimentación DC 24 V	20,4 a 28,8

Figura 3.48 Datos de las entradas del EM235.

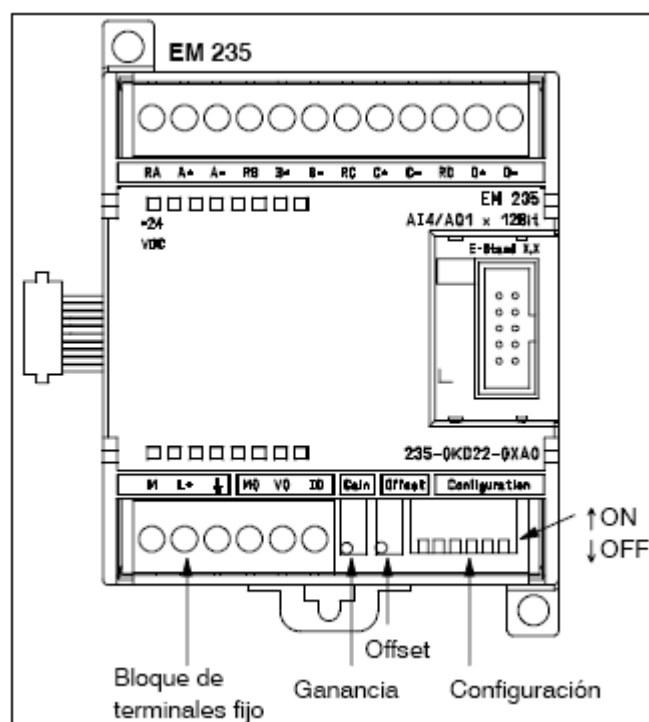


Figura 3.49 Esquema del módulo EM235.

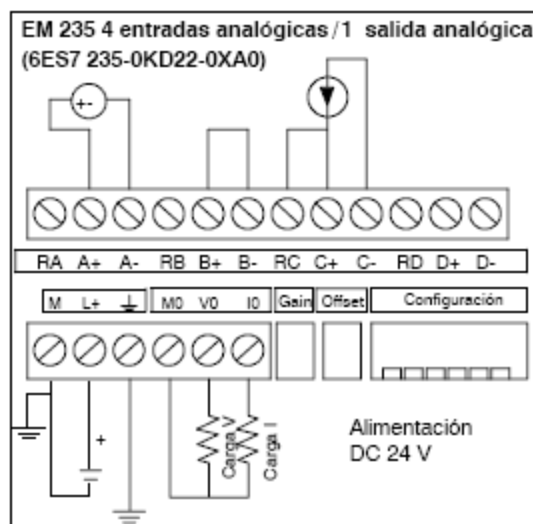


Figura 3.50 Cableado de las entradas y salida analógica del EM235.

La configuración del módulo de ampliación EM235 se realiza utilizando los interruptores DIP. El rango de las entradas analógicas y la resolución se seleccionan con los interruptores 1 a 6.

Todas las entradas se activan en un mismo rango y formato. La siguiente figura muestra cómo seleccionar el formato unipolar/bipolar (interruptor 6), la ganancia (interruptores 4 y 5) y la atenuación (interruptores 1, 2 y 3).

Unipolar						Rango de tensión	Resolución
Int1	Int2	Int3	Int4	Int5	Int6		
ON	OFF	OFF	ON	OFF	ON	0 a 50 mV	12,5 μ V
OFF	ON	OFF	ON	OFF	ON	0 a 100 mV	25 μ V
ON	OFF	OFF	OFF	ON	ON	0 a 500 mV	125 μ V
OFF	ON	OFF	OFF	ON	ON	0 a 1 V	250 μ V
ON	OFF	OFF	OFF	OFF	ON	0 a 5 V	1,25 mV
ON	OFF	OFF	OFF	OFF	ON	0 a 20 mA	5 μ A
OFF	ON	OFF	OFF	OFF	ON	0 a 10 V	2,5 mV
Bipolar						Rango de tensión	Resolución
Int1	Int2	Int3	Int4	Int5	Int6		
ON	OFF	OFF	ON	OFF	OFF	\pm 25 mV	12,5 μ V
OFF	ON	OFF	ON	OFF	OFF	\pm 50 mV	25 μ V
OFF	OFF	ON	ON	OFF	OFF	\pm 100 mV	50 μ V
ON	OFF	OFF	OFF	ON	OFF	\pm 250 mV	125 μ V
OFF	ON	OFF	OFF	ON	OFF	\pm 500 mV	250 μ V
OFF	OFF	ON	OFF	ON	OFF	\pm 1 V	500 μ V
ON	OFF	OFF	OFF	OFF	OFF	\pm 2,5 V	1,25 mV
OFF	ON	OFF	OFF	OFF	OFF	\pm 5 V	2,5 mV
OFF	OFF	ON	OFF	OFF	OFF	\pm 10 V	5 mV

Figura 3.51 Configuración de la resolución y rango de tensión en el EM235.

Interruptores de configuración						Unipolar bipolar	Ganancia	Atenuación
Int1	Int2	Int3	Int4	Int5	Int6			
					ON	Unipolar		
					OFF	Bipolar		
			OFF	OFF			x1	
			OFF	ON			x10	
			ON	OFF			x100	
			ON	ON			No válido	
ON	OFF	OFF						0,8
OFF	ON	OFF						0,4
OFF	OFF	ON						0,2

Figura 3.52 Configuración de la ganancia y atenuación en el EM235.

3.5.2. CALIBRACIÓN DE LAS ENTRADAS.

Los ajustes de calibración afectan a la fase de amplificación de la instrumentación que sigue al multiplexor analógico.

Por consiguiente la calibración afecta a todos los canales de entrada de usuario. Cualquier variación de los valores de los circuitos de entrada que preceden al multiplexor analógico provocará diferencias mínimas entre los valores de los distintos canales que estén conectados a la misma señal, incluso después de la calibración.

Con objeto de cumplir las especificaciones es preciso utilizar filtros de entrada para todas las entradas analógicas del módulo, 64 o más muestreos serán necesarios para calcular el valor promedio.

Para calibrar una entrada, se deben seguir los pasos siguientes:

1. Desconectar la alimentación del módulo. Seleccionar el rango de entrada deseado.
2. Conectar la alimentación de la CPU y del módulo. Esperar 15 minutos para que el módulo pueda estabilizarse.
3. Mediante una fuente de tensión o de intensidad, aplicar una señal de valor cero a una de las entradas.
4. Leer el valor que la CPU ha recibido del correspondiente canal de entrada.
5. Con el potenciómetro OFFSET, seleccionar el valor cero u otro valor digital.
6. Aplicar una señal de rango máximo a una entrada. Leer el valor que ha recibido la CPU.
7. Con el potenciómetro GAIN, seleccionar el valor 32000 u otro valor digital.

8. En caso necesario, se debe volver a calibrar el desplazamiento de los potenciómetros (OFFSET) y la ganancia (GAIN).⁶⁷

3.6. PROTOTIPO DE PRUEBA.

El prototipo de prueba consta de un tablero en donde se encuentra montado el PLC y el módulo de extensión analógico EM235 con sus respectivas entradas y salidas, además constituye un hardware adicional en donde se colocan los sensores y actuadores y un visualizador de los datos por medio de LCDs y Microcontroladores PICs.

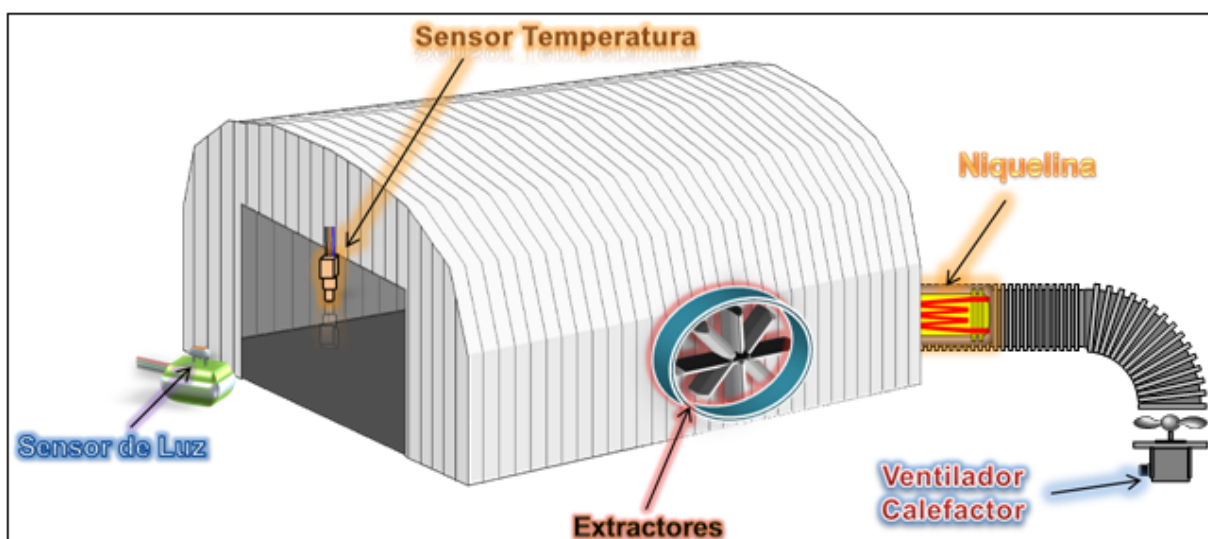


Figura 3.53 Hardware adicional para los sensores y actuadores.

⁶⁷ Toda la información referente al módulo de extensión analógico EM235, se encuentra en el Manual de usuario SIEMENS Simatic de Step7 versión 3: “manual S7200_v3.pdf”, con el título: “Sistema de automatización S7-200 Manual del Sistema”, adquirido con el software instalador de Step7 Micro Win 32 v4.

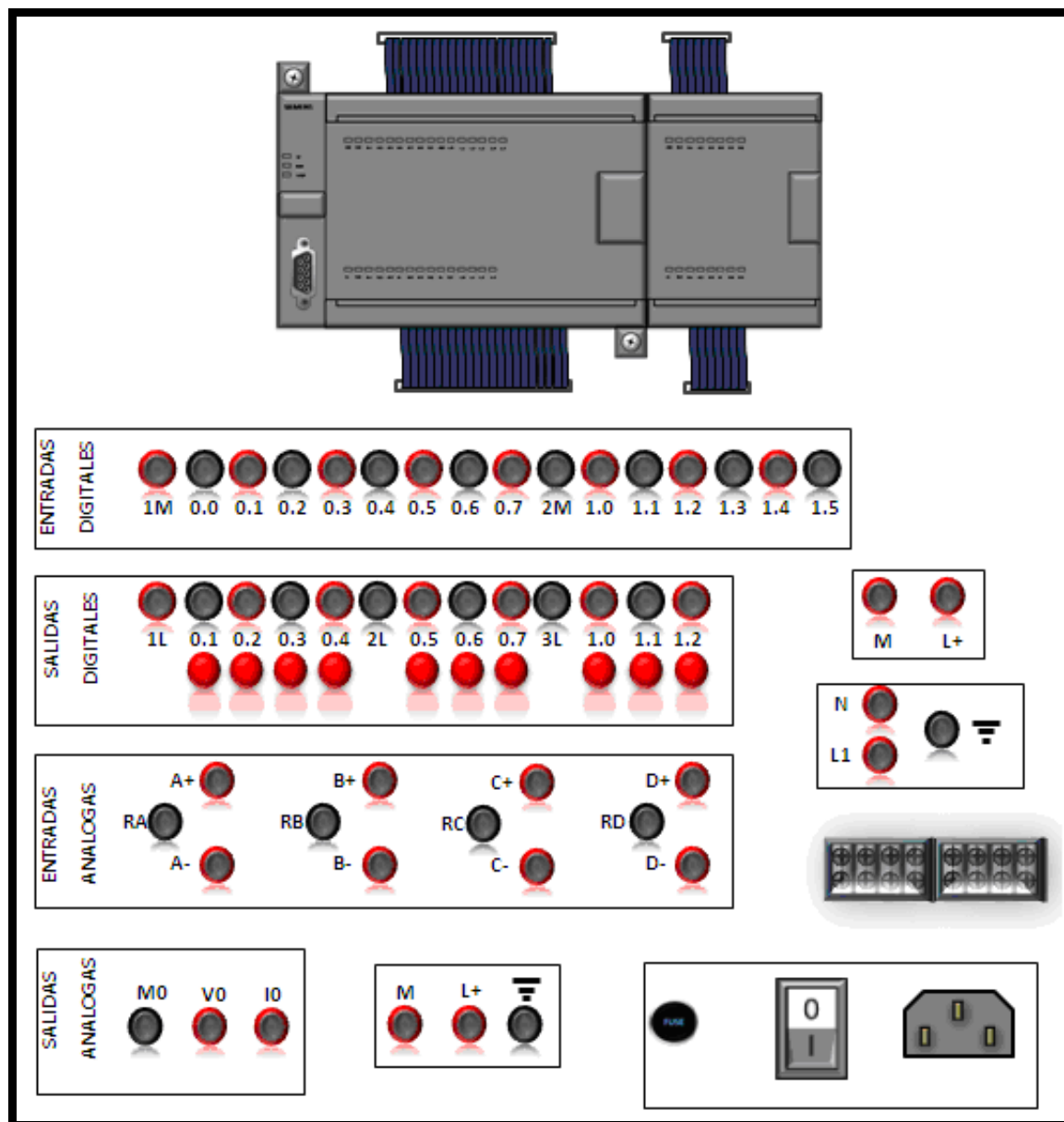


Figura 3.54 Tablero con montaje de PLC y EM235.

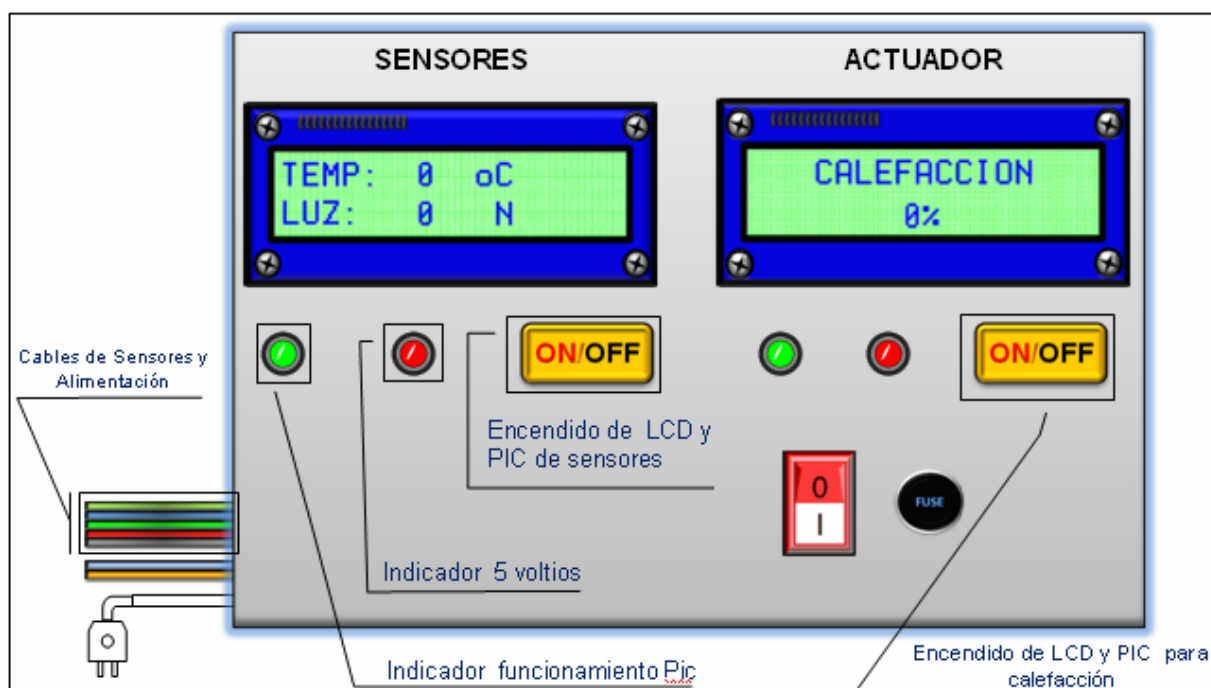


Figura 3.55 Visualizador de datos.

CAPÍTULO 4. SOFTWARE

El proyecto de tesis consta de 3 herramientas de programación que son: LabView, Lookout y Step 7; debido a que los sistemas HMI/Scada están constituidos de hardware y software tanto para la adquisición, control y supervisión se debe establecer con claridad el objetivo de cada lenguaje.

Para la adquisición de datos se utiliza LabView, para el control se recurre a Step 7 y finalmente la supervisión se realiza en Lookout.

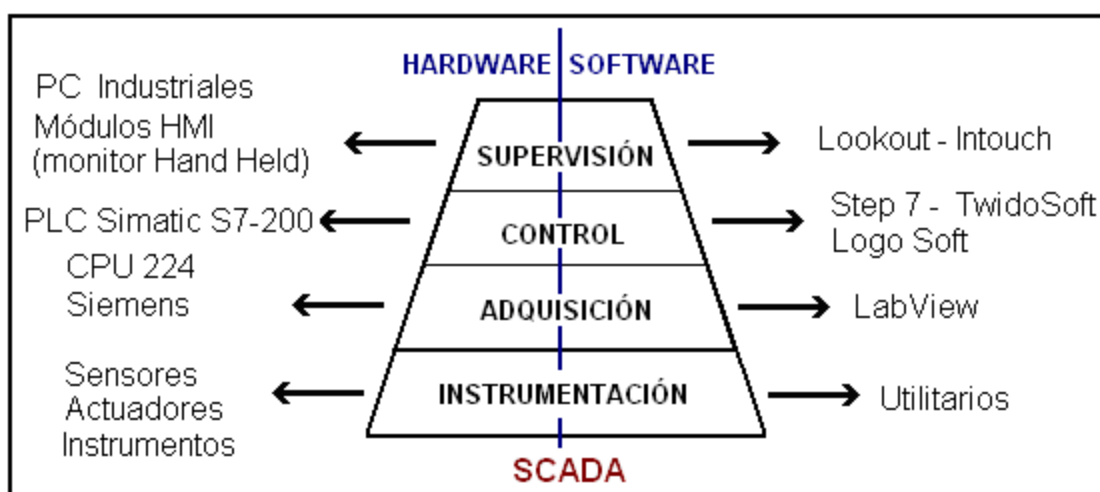


Figura 4.1 Hardware y Software de un Scada.

Debido a que se implementa lógica difusa en el proyecto, se debe programar todo el proceso en LabView como también en Step 7, porque tenemos 2 situaciones siguientes:

- Cuando se está realizando la supervisión del sistema, es decir, en laboratorio mediante Lookout, se necesita tener el proceso difuso en el PC "en LabView" y establecer comunicación con el PLC, por medio de Step 7.
- Cuando el autómatas trabaja en campo, necesita el proceso de lógica difusa programado en Step 7 y grabado en su memoria, es decir cuando no hay supervisión del sistema; pero el operador en cualquier momento puede acceder a él y comprobar sus datos y proceso conectándose con una PC portátil y cargando una interfaz LabView y Lookout.

4.1 DIAGRAMAS.

UML significa Lenguaje de Modelamiento Unificado, es una herramienta importante para entender el desarrollo del software. No es un lenguaje de programación, más bien es un lenguaje de información que nos ayuda a documentar la organización de los sistemas y en el diseño de los mismos.

UML se utiliza para modelar sistemas de programación orientada a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos⁶⁸.

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con UML, constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Dentro de las dos metodologías “UML y RUP” se detallan diagramas similares tales como: Caso de Uso, Clases, Secuencia, Estados.

Los 13 tipos⁶⁹ de diagramas UML se muestran en la figura 4.2.

Los diagramas de Estructura destacan los elementos que deben existir en el sistema modelado, los diagramas de Comportamiento enfatizan lo que debe suceder en el sistema y los diagramas de Interacción son un subtipo de diagramas de comportamiento, que despliegan el flujo de control y datos entre los elementos de la aplicación.

En el presente trabajo se ha determinado realizar sólo los diagramas que se pueden utilizar en programación estructurada y diagramas esquemáticos para la conexión hardware - software en sistemas de la industria en general ya que algunos diagramas se utilizan estrictamente para la programación orientada a objetos (diagramas de clases, diagramas de objetos, diagramas de estructura compuesta estática y dinámica, etc.)

⁶⁸ <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>

⁶⁹ http://www.vico.org/aRecursosPrivats/TRAD_introUML.pdf

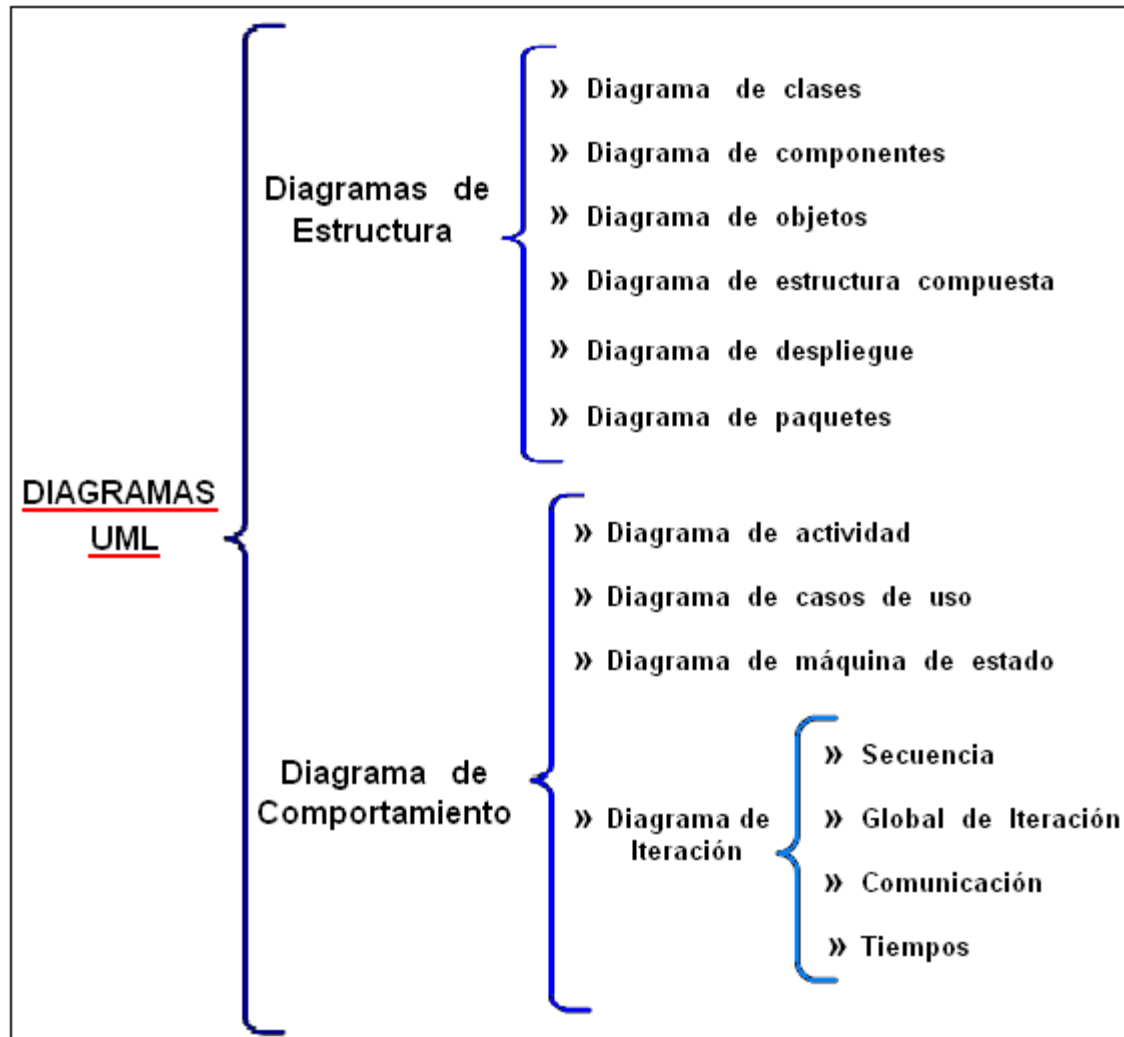


Figura 4.2 Diagramas UML.

4.1.1 DIAGRAMA DE COMPONENTES Y DISTRIBUCIÓN.

Un componente de software es una parte física de un sistema, y se encuentra en la computadora, no en la mente del analista.

Ejemplos de componentes son tablas, archivos de datos, ejecutables, bibliotecas de vínculos dinámicos, documentos, ventanas, etc.⁷⁰

Lo que contiene este diagrama son interfaces y relaciones, aunque también pueden aparecer otros tipos de símbolos.

⁷⁰ <http://www.dsi.uclm.es/assignaturas/42530/pdf/M2tema12.pdf>

El símbolo principal de un diagrama de componentes es un rectángulo que tiene otros dos sobrepuestos en su lado izquierdo, con el nombre del componente dentro del rectángulo más grande por ejemplo:



Figura 4.3 Ejemplo de un componente.

Las relaciones de dependencia entre componentes se dibujan con líneas entrecortadas y flechas.

Los diagramas de distribución se enfocan específicamente al hardware de un sistema determinado, el elemento primordial del hardware es un *nodo*, que es un nombre genérico para todo tipo de recurso de cómputo y se representa mediante un cubo, dentro del cubo se puede introducir información sobre el nodo, que puede ser simplemente texto o inclusive componentes, usando los diagramas de componentes anteriormente señalados.

Los diagramas de distribución se encargan de dibujar hardware, pero se los puede detallar con los diagramas de componentes en su interior, cada uno de los nodos puede contener otros componentes, incluyendo software, lo cual debe ser especificado. Es decir se pueden combinar los diagramas de Componentes y de Distribución.

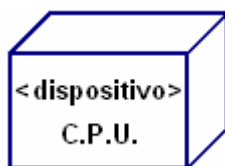


Figura 4.4 Ejemplo de un nodo.

En el presente proyecto se realizan dos diagramas de distribución y componentes, el primero estableciendo que el proceso de lógica difusa se realiza en LabView y el segundo desarrollando lógica difusa en Step7:

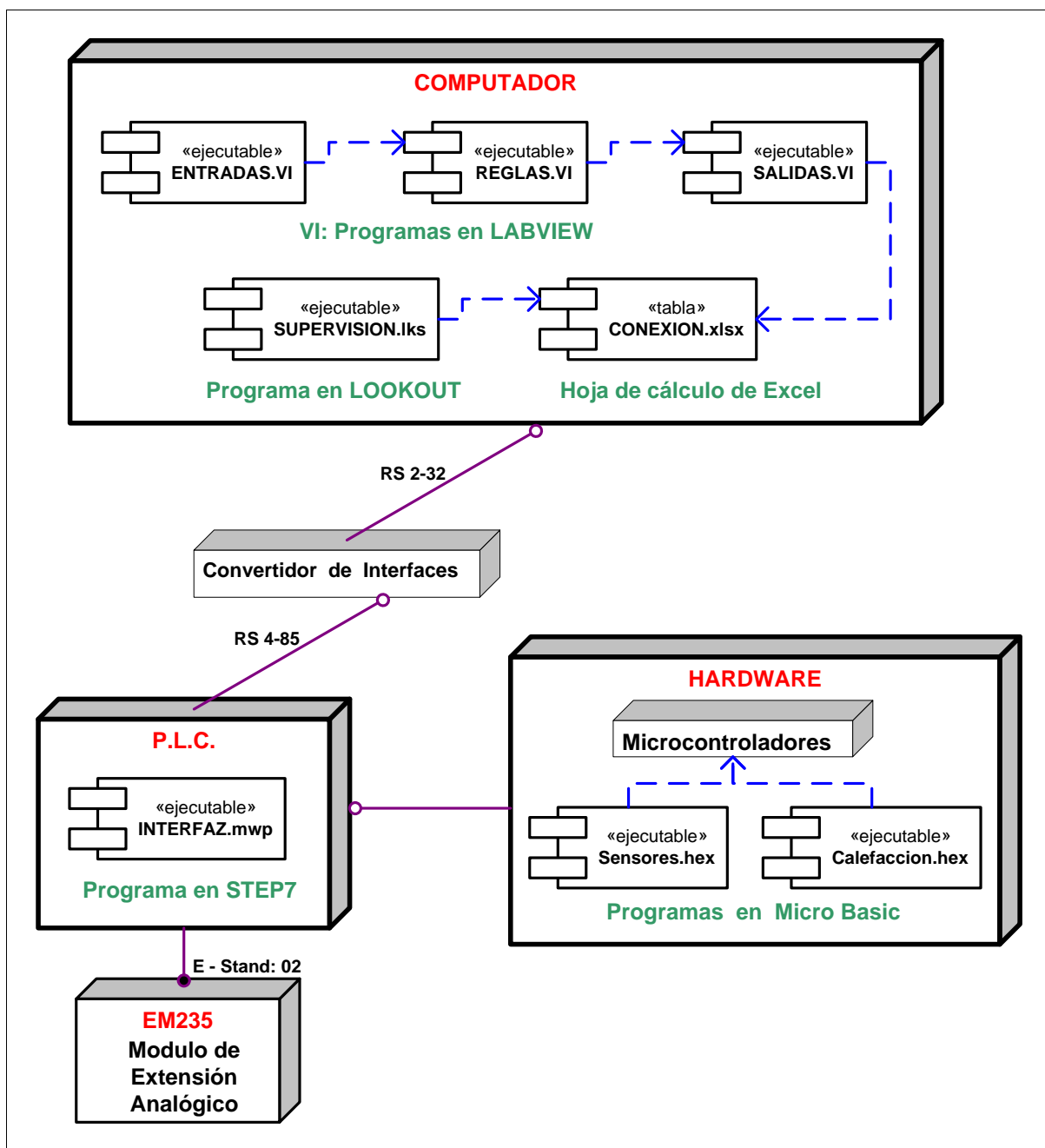


Figura 4.5 Diagrama de Componentes y Distribución 1.

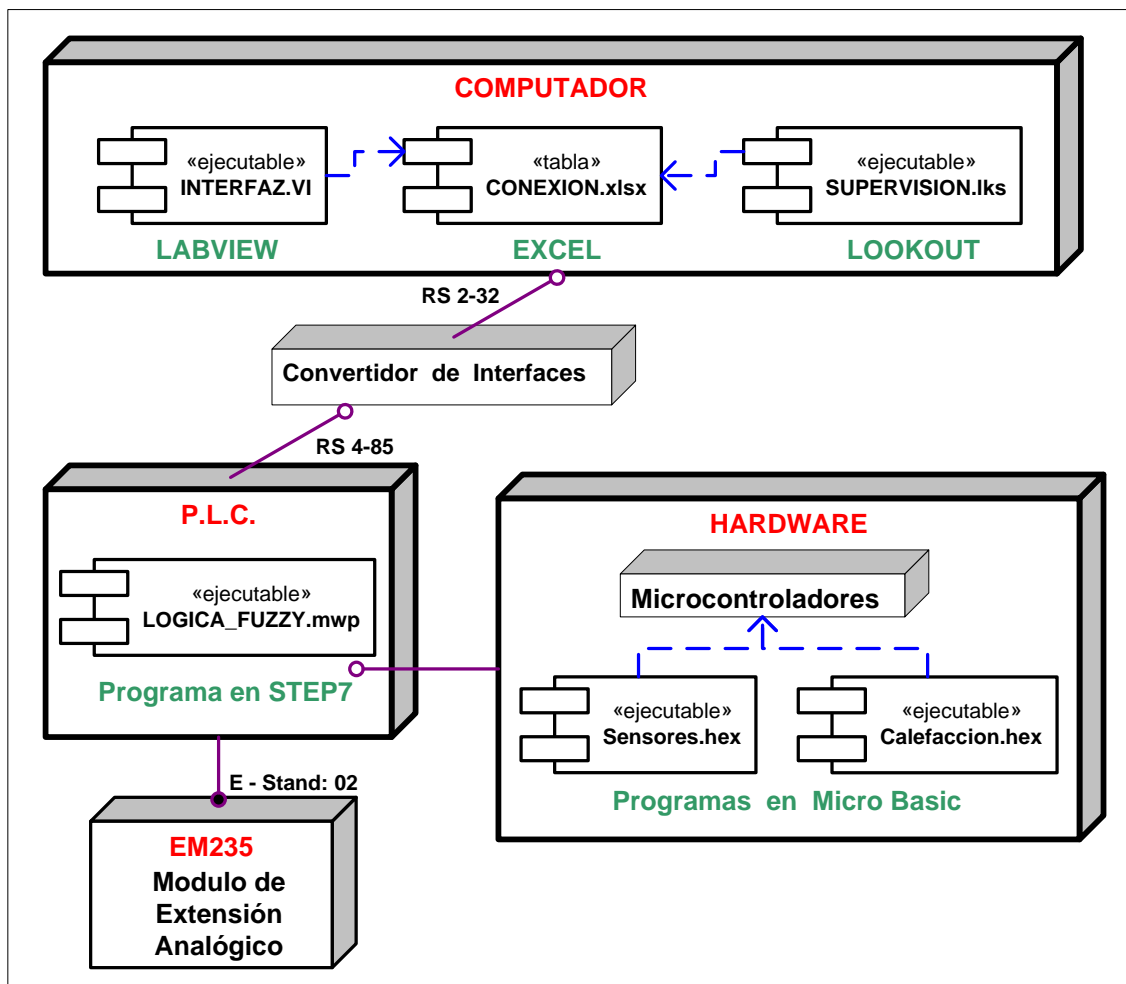


Figura 4.6 Diagrama de Componentes y Distribución 2.

4.1.2 DIAGRAMA DE ACTIVIDADES.

El diagrama de actividades es muy parecido a los viejos diagramas de flujo, pues muestra los pasos (conocidos como *actividades*) así como puntos de decisión y bifurcaciones. Lo que hacen es mostrar una visión simplificada de lo que ocurre durante una operación o proceso.

Cada actividad se representa por un rectángulo con las esquinas redondeadas. Una flecha representa la transición de una a otra actividad. Cuenta con un punto inicial (representado por un círculo relleno) y uno final (representado por una diana).

A continuación el diagrama de actividades del presente proyecto:

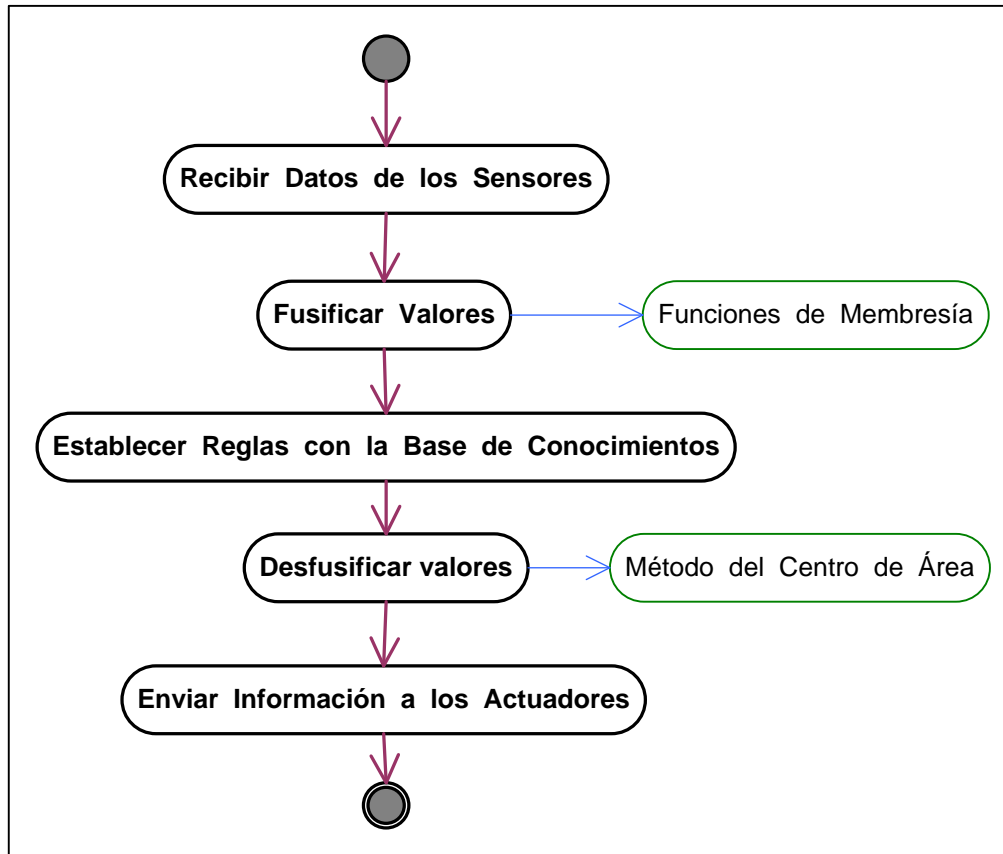


Figura 4.7 Diagrama de Actividades.

4.1.3 DIAGRAMA DE ESTADO.

Los diagramas de estado surgen por cambios en el sistema, debido a que los objetos o componentes que forman dicho sistema modificaron su *estado* como respuesta a los sucesos y al tiempo.

El icono para el estado es un rectángulo y el símbolo de una transición es una línea continua y una punta de flecha. El círculo relleno se interpreta como el punto inicial de una secuencia de estados y la diana representa al punto final.

Se puede subdividir el estado en 3 áreas que muestren: el nombre, variables y actividades, dentro de las actividades se detallan:

- Entrada: Lo que sucede cuando el sistema entra al estado.
- Salida: Lo que ocurre cuando el sistema sale del estado.
- Hacer: Lo que pasa cuando el sistema está dentro del estado.
- Se puede agregar otros conforme sea necesario.

El proyecto consta de dos diagramas de estado, el primero para lógica difusa en LabView y el segundo para lógica difusa en Step7:

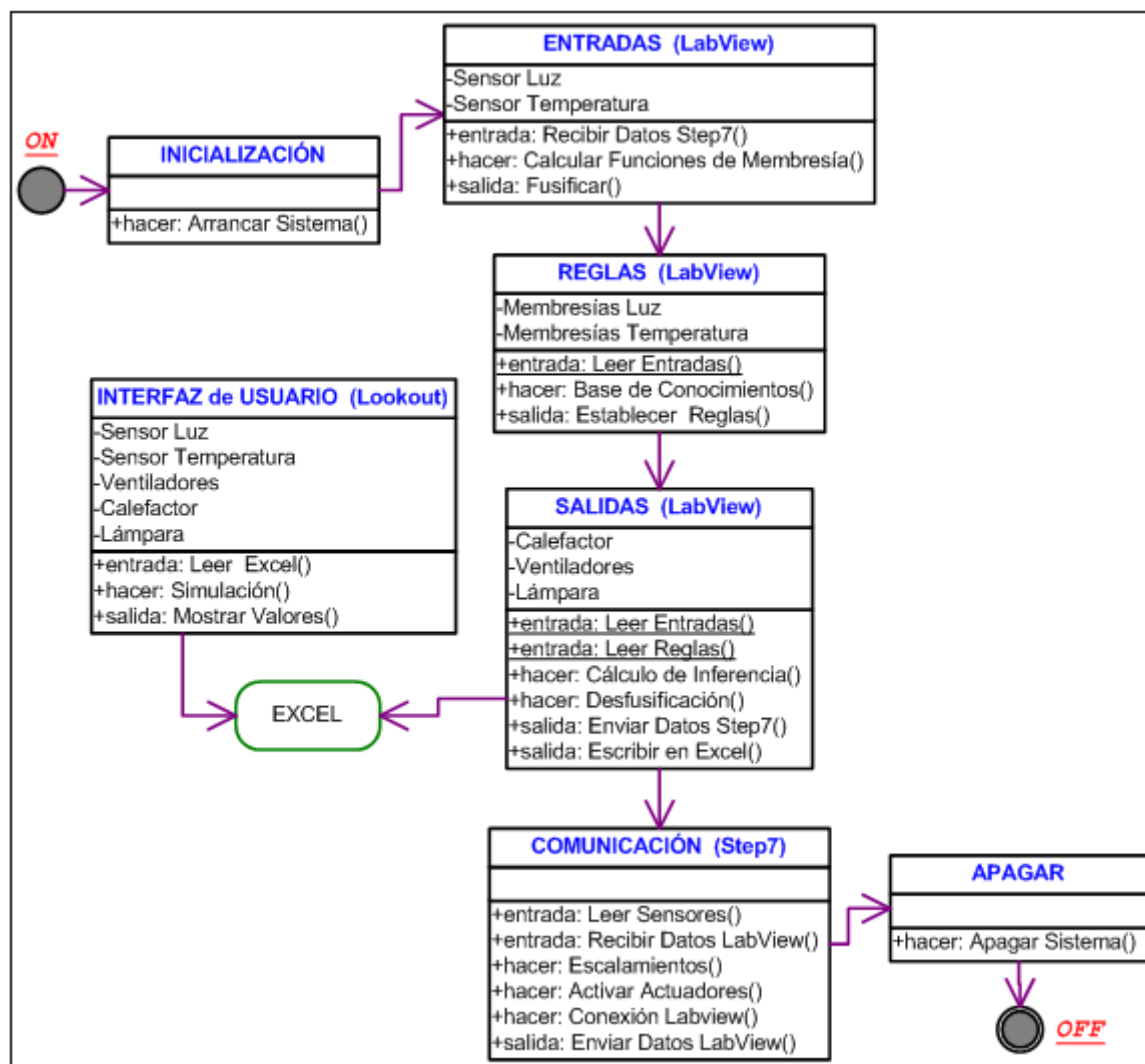


Figura 4.8 Diagrama de Estados 1.

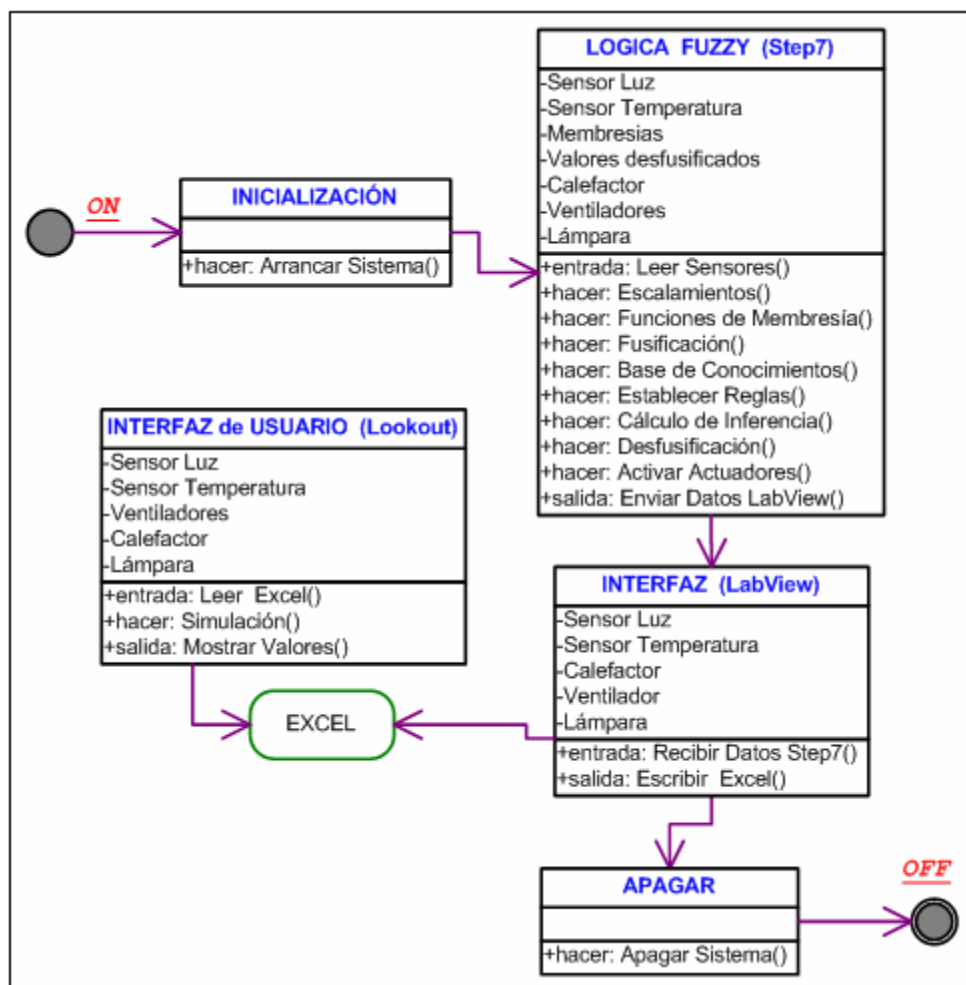


Figura 4.9 Diagrama de Estados 2.

4.1.4 DIAGRAMA DE SECUENCIA.

Este tipo de diagramas muestra una interacción ordenada según la secuencia de eventos con perspectiva a una línea de tiempo, se muestran los objetos participantes en la interacción y los mensajes ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo y en el eje horizontal se ubican los objetos y actores participantes en la interacción colocándolos de izquierda a derecha y en la parte superior.

Cada objeto o actor tiene una línea vertical y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo. Se pueden colocar etiquetas junto a las transiciones o activaciones a las que se refieren.

La línea de vida de un objeto es discontinua y se desplaza hacia abajo. Una línea continua con una punta de flecha conecta a una línea de vida con otra, representa un mensaje de un objeto a otro.

El tiempo se inicia en la parte superior y continúa hacia abajo. No necesariamente se debe especificar el tiempo de manera explícita (como en segundos, minutos, horas, días, etc.), aunque se puede hacer si resulta necesario. Siempre se tiene que ubicar correctamente en el eje vertical la secuencia correcta de eventos que se deben ir dando de forma cronológica o en una línea de tiempo.

En este caso también se debe realizar dos diagramas de secuencia correspondientes al proyecto el primero con lógica difusa en LabView y el segundo con lógica difusa en Step7:

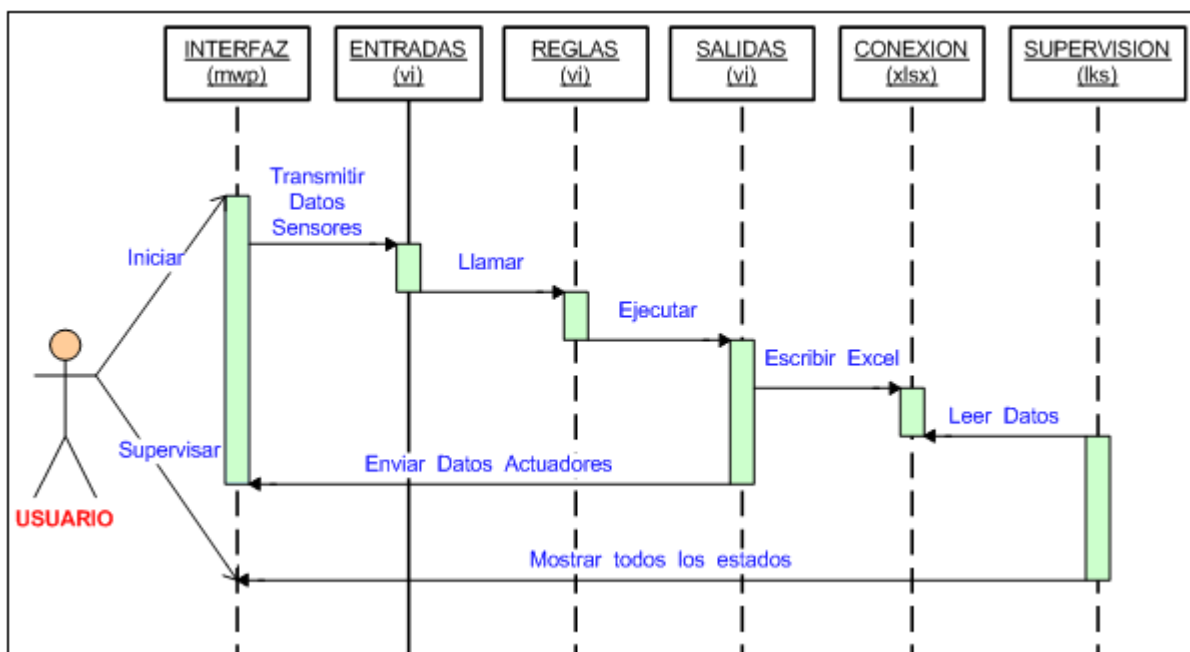


Figura 4.10 Diagrama de Secuencia 1.

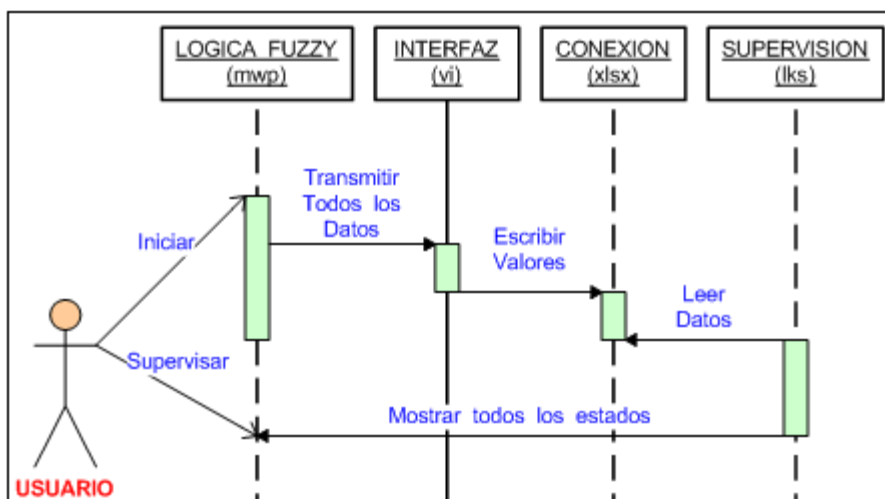


Figura 4.11 Diagrama de Secuencia 2.

4.2 LOOKOUT.

Lookout es un software HMI/Scada, creado por la empresa desarrolladora National Instruments, para el monitoreo y supervisión de procesos industriales.

Es un poderoso software de automatización industrial, corre bajo Windows, y se comunica con dispositivos de campo I/O, con los Controladores Lógico Programables “PLCs”, con las Unidades Terminales Remotas “RTUs”, y otros dispositivos. Los proyectos de Lookout incluyen: proceso continuo, fabricación discreta, aplicaciones de lote y sistemas de telemetría remotos.

Es orientado a objetos y manejado por eventos.⁷¹

Lookout tiene muchas capacidades diversas como el Mando del Proceso Estadístico (SPC), recetas, búsqueda SQL, seguridad incorporada, varios procesos múltiples corriendo en una computadora, la animación sofisticada, alarmas complejas, telemetría, radio y dial, senderos de auditoría de eventos, ajustes de set points, apoyo de multimedia, compatibilidad con pantallas touch screen, conexión en red de computadoras, Intercambio Dinámico de Datos (DDE⁷² y NetDDE), y más.

Con Lookout se puede desarrollar una aplicación completamente en línea, sin cerrar, no se necesita compilar o transmitir a una base de datos cada vez que se hace una modificación o ejecutar el programa de desarrollo y el de edición por

⁷¹ Introducción a Lookout 5.0.- Manual de Usuario. (Ayuda de programa). National Instruments (ni.com).

⁷² DDE “Exchange Dinamic Data” Intercambio Dinámico de Datos.

separado, no es necesario interrumpir el proceso o que los dispositivos de campo paren de trabajar.

Se puede utilizar Lookout con otros programas de Microsoft Windows multitarea, por ejemplo con Excel.

Es un software para manejo en tiempo real, es modular y todo elemento es un objeto.

Los objetos poseen: Funcionalidad, Parámetros o Propiedades y Bases de Datos.

La funcionalidad se refiere al objetivo de creación del objeto, labor que debe cumplir. Los parámetros o propiedades son los atributos de los objetos para que sean manipulados. La base de datos guarda los posibles valores que puede tomar el objeto.

Al guardar un proyecto de Lookout se generan 4 archivos:

- l4p: Archivo de Proceso.- Es el archivo compilado que contiene el proceso, se actualiza cada vez que se selecciona guardar, es un archivo binario.
- lks: Código Fuente.- Es el archivo que guarda el código fuente y puede abrirse y editarse en un bloc de notas porque están código ASCII, incluye las definiciones de los objetos, nombres, configuraciones I/O, comunicaciones, lógica de mando, diseño del panel de control, etc. Este archivo también puede ser recompilado para abrirlo en otras versiones.
- l4t: Archivo de Estado.- Contiene los valores reales de miembros de datos de cada objeto, estos valores incluyen los set points. El archivo de proceso utiliza los valores del archivo de estado para guardar las posiciones de los objetos, también pone al día la base de datos histórica.
- lka: Archivo de Seguridad.- Toda la información de seguridad para un proceso está aquí, sin este archivo todos los usuarios tienen acceso completo a todas las partes del proceso porque no existen los permisos y niveles de seguridad.

ARQUITECTURA:

Lookout es un software basado en eventos que admite permanecer en línea durante la configuración y edición de los objetos. No requiere de compilación, trabaja con una base de datos no centralizada es decir distribuida y los tiempos de respuesta no se degradan.

Orientado a objetos.- Todos los objetos son tratados por igual, si se necesita un objeto que Lookout no lo tenga, se lo puede crear (un archivo *.cbl) con su .dll, ésta técnica aísla capacidades y funcionalidades de objetos por objetos, con su base de datos distribuida no es necesario interrogarla continuamente para verificar los valores que cambian.

Basado en eventos.- Un evento es un cambio en el valor de un dato, cuando una señal de entrada cambia, el objeto despierta para procesar el valor en su funcionalidad, los objetos mandan señales, sólo cuando cambian de resultado y funcionalidad, causando así un evento.

FUNCIONAMIENTO:

En Lookout no existe código fuente, como en los típicos lenguajes de programación, los objetos creados se relacionan entre ellos mediante la edición de conexiones y se realizan las respectivas configuraciones.

Por ejemplo, para crear un Switch se realizan las siguientes acciones:

Primero creamos un nuevo proceso, luego creamos un panel o ventana en la que se va a trabajar, después hacemos clic en: object => create, aparece una ventana en donde el usuario puede escoger el objeto Lookout o ActiveX a utilizar, buscamos un Switch, luego seleccionamos el proceso en el que se trabaja, seguidamente aparece una ventana de propiedades del objeto (nombre, estado, mando local o remoto, etc), y por último una mini ventana indicando un gráfico para asociar al objeto, pudiendo seleccionar uno de los indicados o buscando en la carpeta de Gráficos de Lookout.

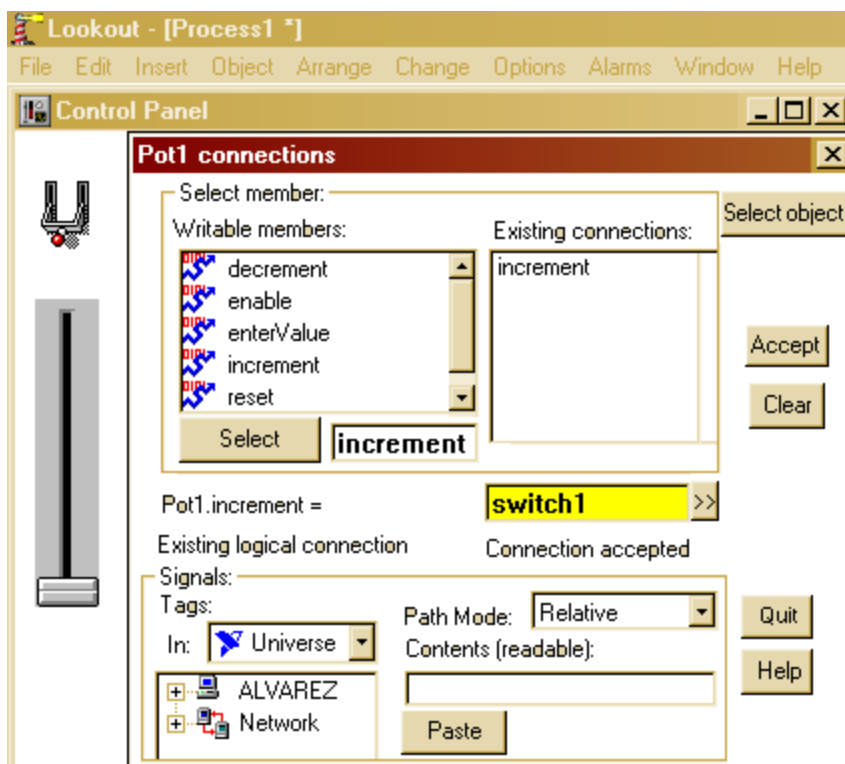


Figura 4.12 Edición de conexiones entre objetos.

Para crear un potenciómetro seguimos los mismos pasos. Si deseamos que el potenciómetro se incremente automáticamente, cuando el switch esté en ON, debemos ir a: Object => Edit Connections, luego seleccionamos el objeto a ser controlado en este caso el potenciómetro, seguidamente aparecerá la ventana de conexiones, seleccionamos “increment” y colocamos la orden Switch1 (nombre del objeto que controla). Por último con las teclas: Ctrl + barra espaciadora, podemos pasar de modo edición a modo ejecución y viceversa, inmediatamente comienza a funcionar.

De este modo podemos realizar el control de todo un programa, además si es necesario se pueden crear objetos expresión de tipo: NIF si la expresión es numérica, LIF si es lógica y TIF si es textual. Haciendo clic en object => create => (expresion). También se pueden crear expresiones de objetos existentes, en: insert => Expresión... y seleccionamos el objeto al que necesitamos sacarle una expresión.

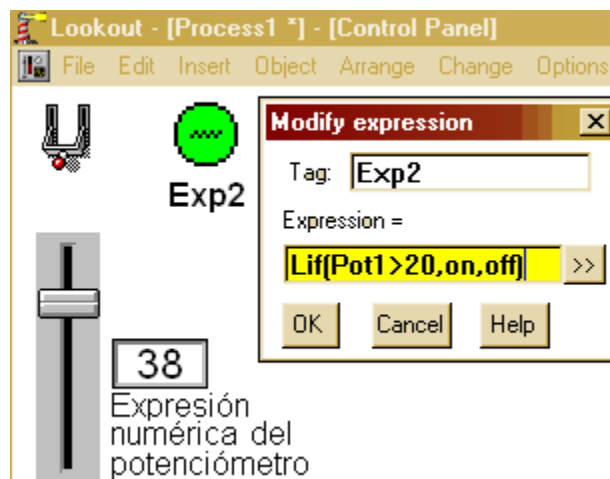


Figura 4.13 Expresión de un objeto y objeto expresión.

Se debe tomar en cuenta que un objeto expresión es diferente a la expresión de un objeto, ya que el objeto expresión trabaja independientemente y es otro elemento del conjunto, en contraste con la expresión de un objeto que es una forma más de representar a un objeto, porque un objeto puede tener varias expresiones, por ejemplo el potenciómetro tiene su expresión gráfica y su expresión numérica y podría también tener otras.

En Lookout se pueden crear animaciones mediante: object => create => animator. Mediante este objeto podemos crear cualquier tipo de animación con un diseño gráfico repetido varias veces seguidas en diferentes estados, por ejemplo el siguiente caballo tiene 5 estados cabalgando, dibujado en 5 filas y 1 columna, en Lookout parece que está corriendo, si configuramos: en X una variable que aumente su valor, en Cel/Rate la velocidad de cambio entre estados, en Rows el número de filas y en Columns el número de columnas que tenga el gráfico en total.

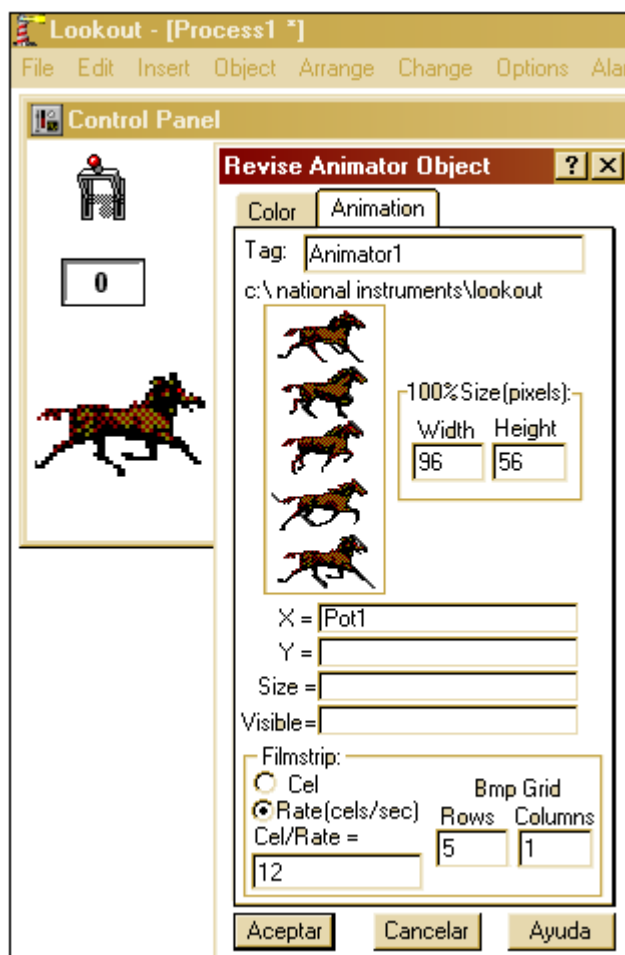


Figura 4.14 Objeto animator.

Dentro de los objetos que posee Lookout, tenemos: retardos o delays, timers, contadores, alarmas, objetos manejadores de bases de datos, latch gates, hiper trends históricos, objeto modbus, paneles, potenciómetros, switches, pulsos o generadores de pulsos, recetas o recetas, reportes, botones, escalas, spinners, relojes, waveforms, tablas, indicadores, bombas, ventiladores, válvulas, mezcladoras, cisternas, bandas transportadoras, tubería, objetos de animación, etc, etc, etc, además muchos gráficos que se pueden asociar a cualquier objeto y por último contiene numerosos objetos ActiveX.

4.2.1 INTERFAZ DE USUARIO.

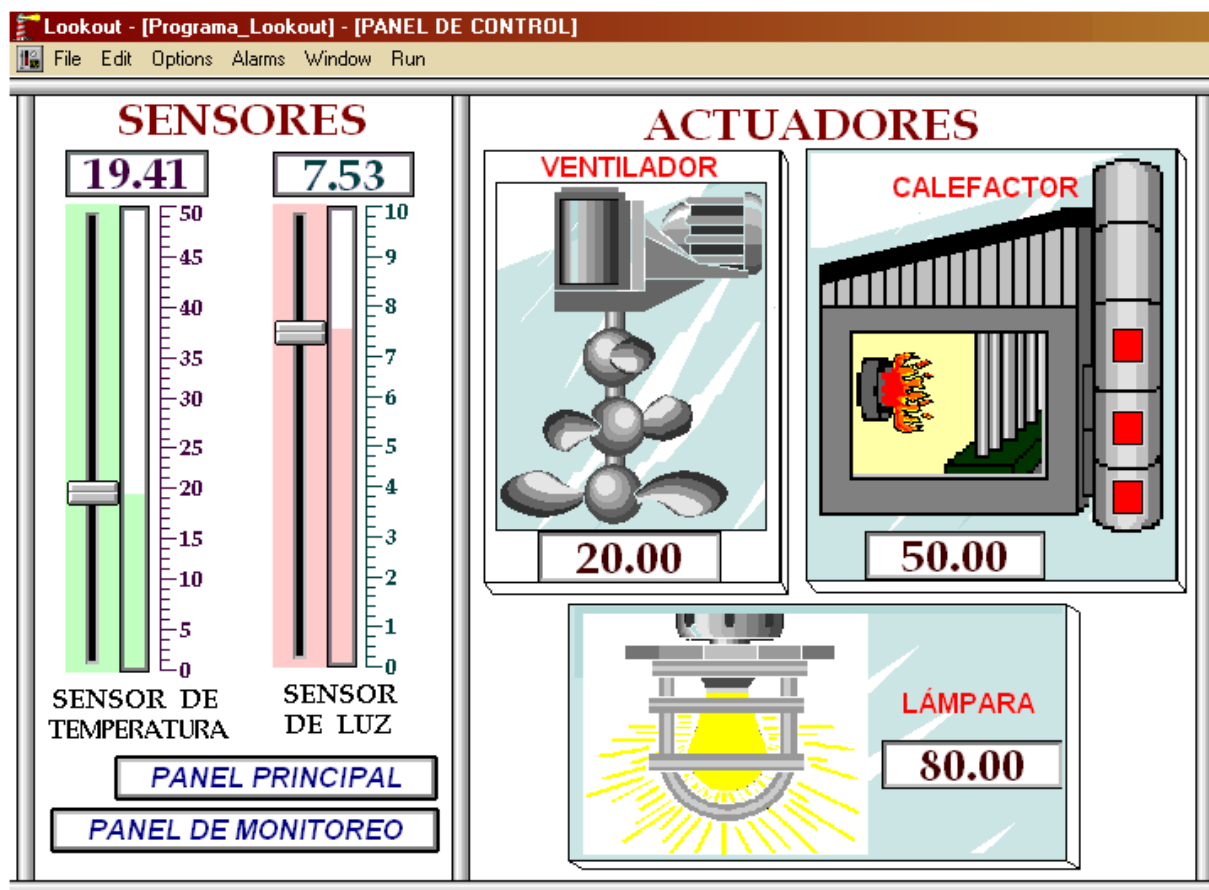


Figura 4.15 Interfaz Lookout. Panel de Control.

El programa también cuenta con la pantalla de monitoreo de señales mediante el hipertrend histórico. Además cuenta con animación en todo el proceso.

Lo más importante es la conexión con Microsoft Excel mediante el objeto DdeLink (DDE), con los parámetros correspondientes para leer los datos desde Excel que le son entregados por LabView.

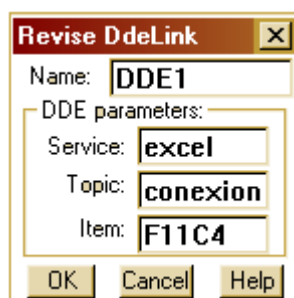


Figura 4.16 Objeto DdeLink.

4.3 LABVIEW.

LabView es un software de desarrollo, es una herramienta diseñada especialmente para controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos, puertos serie y GPIBs (Buses de Intercambio de Propósito General).

Es un lenguaje de programación de propósito general, como es el Lenguaje C o Basic, pero con la característica que es totalmente gráfico, facilitando de esta manera el entendimiento y manejo de dicho lenguaje para el diseñador y programador de aplicaciones tipo HMI/Scada.

Incluye librerías para la adquisición, análisis, presentación y almacenamiento de datos, GPIB y puertos serie. Además de otras prestaciones, como la conectividad con otros programas, por ejemplo de cálculo y en especial MatLab.

Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o sub-módulos mucho más sencillos. Además estos módulos pueden ser usados en otras tareas, con lo cual permite una programación más rápida y provechosa.

También ofrece la ventaja de "debugging" en cualquier punto de la aplicación. Permite la posibilidad de poner "break points", ejecución paso a paso, ejecución hasta un punto determinado y se puede observar como los datos van tomando valores a medida que se va ejecutando la aplicación. Además también lleva incorporado generadores de señales para poder hacer un simulador.⁷³

FUNCIONAMIENTO:

LabView es un lenguaje completamente gráfico, y el resultado de ello es que es totalmente parecido a un instrumento, por ello a todos los módulos creados con LabView se les llama VI "Virtual Instrument" (Instrumento Virtual).

Existen dos conceptos básicos en LabView: el Front Panel (Panel Frontal) y el Block diagram (Diagrama de Bloques). El Panel Frontal es la interfaz que el usuario está viendo y puede ser totalmente parecido al instrumento del cual se están recogiendo los datos, de esta manera el usuario sabe de manera precisa cual es el estado actual de dicho instrumento y los valores de las señales que se están midiendo. El diagrama de bloques es el conexionado de todos los controles

⁷³ <http://www.highlights.com.ec/docs/InstCont04/401904.pdf> ¿Qué es LabView?

y variables, que tendría cierto parecido al diagrama del esquema eléctrico del instrumento.

LabView tiene la característica de descomposición modular ya que cualquier VI que se ha diseñado puede convertirse fácilmente en un módulo que puede ser usado como una sub-unidad dentro de otro VI. Esta peculiaridad podría compararse a la característica de procedimiento en los lenguajes de programación estructurada.

Es un sistema abierto, cualquier fabricante de tarjetas de adquisición de datos o instrumentos en general puede proporcionar el driver de su producto en forma de VI dentro del entorno de LabView. También es posible programar módulos para LabView en lenguajes como C y C++, estos módulos son conocidos como Sub-VIs y no se difieren a los VI creados con LabView salvo por la interfaz del lenguaje en el que han sido programados. Además estos Sub-VIs son muy útiles por ejemplo en el campo de cálculos numéricos complejos que no se encuentran incluidos en las librerías de LabView.

Panel Frontal:

Se podría decir que en cualquier VI existen dos caras bien diferenciadas: El Panel Frontal y el Diagrama de Bloques.

El Panel Frontal es la cara que el usuario del sistema está viendo cuando está monitorizando o controlando el sistema, es decir, es la interfaz del usuario. Este contiene controles e indicadores y existe una gran variedad de ellos, pero además incluso se pueden diseñar controles e indicadores personalizados, lo cual permite tener una amplia gama de dichos controles e indicadores.

Un control puede tomar muchas formas y una de estas puede ser el dibujo real, usado en instrumentos reales. Otros son estrictamente conceptos digitales o analógicos.

Pero todos los controles tienen una forma visual que indican al usuario cual es el estado de dicho control en el instrumento real. Es muy importante en un sistema HMI/Scada que el usuario no tenga que interpretar nada, sino que todo le sea claro y conciso, las interpretaciones pueden dar lugar a falsas actuaciones y por consiguiente, podrían existir errores. Además, dos usuarios podrían interpretar de manera diferente cualquier evento.

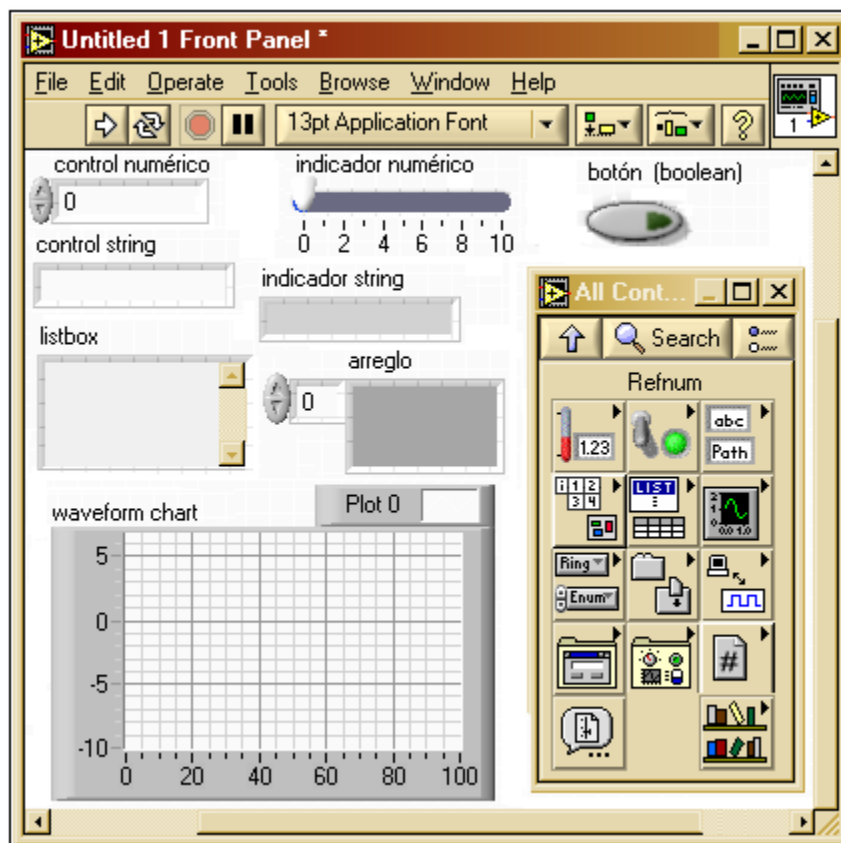


Figura 4.17 Panel Frontal de LabVIEW.

El panel frontal consta de controles e indicadores en su interfaz.

Diagrama de Bloques:

El diagrama de bloques del VI sería la cara oculta del panel frontal, una cara que el usuario del sistema no puede ver. En ella están todos los controles e indicadores interconectados, pareciéndose mucho a un diagrama de esquema eléctrico. Esta cara es mucho menos conceptual que el panel frontal y para el usuario sería muy difícil entenderla.

Todos los módulos están interconectados, mediante líneas de conexión, por donde circulan los diferentes datos o valores del VI, de esta manera se logra que el VI funcione como un conjunto de elementos, módulos y sub-módulos.

La paleta de herramientas se puede utilizar tanto para el diagrama de bloques como para el panel frontal. El diagrama de bloques consta de funciones para realizar diversas operaciones entre los controles e indicadores del panel frontal. La representación de los controles e indicadores en el diagrama de bloques se denominan terminales.

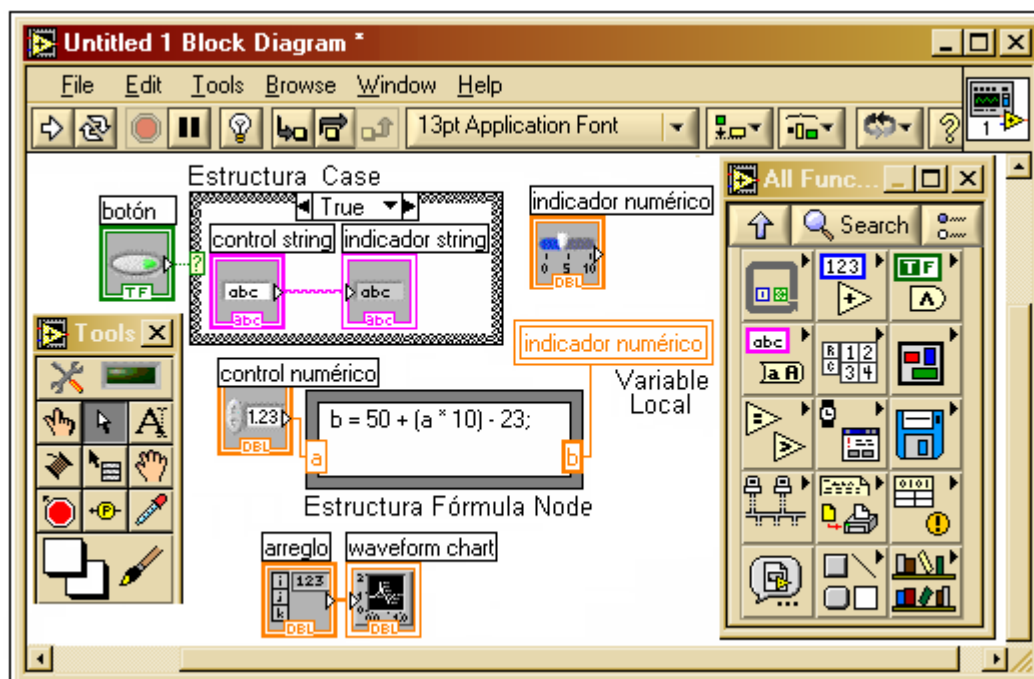


Figura 4.18 Diagrama de Bloques de LabView.

Icono Conector:



Figura 4.19 Icono Conector.

El icono conector nos ayuda a llamar a otros VIs o SubVIs, es la herramienta que ayuda a LabView a ser modular.

El icono conector se encuentra en la parte superior derecha tanto del panel frontal y del diagrama de bloques. En el panel frontal se asocian los controles e indicadores al icono mediante la herramienta conectar, para utilizar el presente VI futuramente en otros programas si es necesario. En el diagrama de bloques es donde se llama a los SubVIs que se necesite en el presente programa.

El icono tiene formas de presentar los VIs, con su configuración puede ser:

- a) Abrir el panel frontal sólo cuando se lo cargue.
- b) Mostrar el panel frontal cuando se lo llame.
- c) Suspender el panel frontal cuando se lo llame.

En esta última opción si se muestra el panel frontal pero no corre.

Para editar el icono se hace un clic derecho sobre él, y se escoge edit icon, al igual para conectar los controles e indicadores en el panel frontal, se elige show connector. Para utilizar otro VI en el diagrama de bloques, en la paleta de herramientas de todas las funciones se escoge select a VI. Y para configurar el nuevo SubVI, se hace clic derecho sobre él, y se seleccionan los parámetros convenientes.

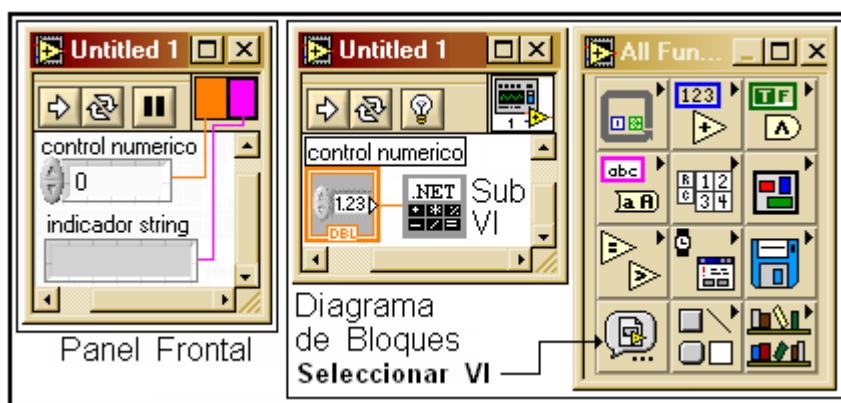


Figura 4.20 Icono conector en panel frontal y diagrama de bloques.

4.3.1 VENTAJAS DE LABVIEW.

LabView brinda varios beneficios en diferentes aspectos como:

- a) Mediciones de alta velocidad de un amplio rango de sensores.⁷⁴
- b) Monitoreo de vibración y mantenimiento predictivo.
- c) Para registro de datos.
- d) Control estadístico de procesos.
- e) Para compartir datos entre PLCs y otros dispositivos de automatización.
- f) Desarrollo de interfaces gráficas de usuario.
- g) Para alarmas, aplicar escalas y reportes.
- h) Conectividad empresarial y con bases de datos.
- i) Algoritmos avanzados y control de movimiento personalizado.

⁷⁴ <http://zone.ni.com/devzone/cda/tut/p/id/6329> Razones para utilizar LabView en un PLC.

- j) Mejor rendimiento y confiabilidad con control por FPGA.
- k) Sus controles e indicadores son precisos, según cada tipo de dato.
- l) Orientado a flujo de datos, permitiendo la observación de su comportamiento.

4.3.2 INTERFAZ DE USUARIO.

Debido a las dos situaciones que se dan tanto en campo como en laboratorio, se debe programar lógica difusa en LabView y en Step 7.

El primer programa LabView constará de todo el proceso fuzzy y el segundo programa solo será una interfaz que se conecte a Lookout y al PLC por medio de comunicación serial.

A continuación se presenta la interfaz del primer programa, que consta de 3 pantallas: entradas, reglas y salida.

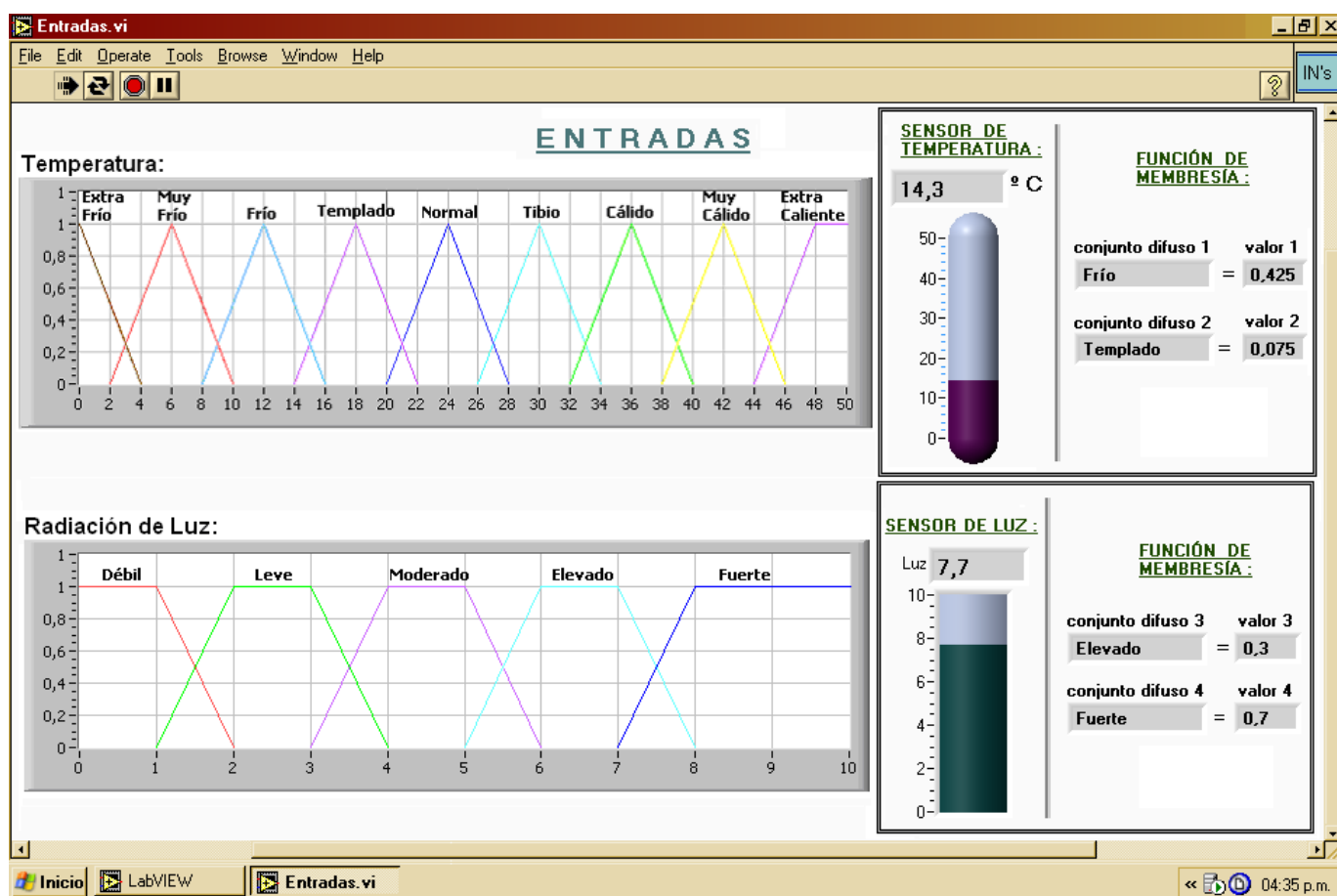


Figura 4.21 Interfaz LabView. Pantalla Entradas.

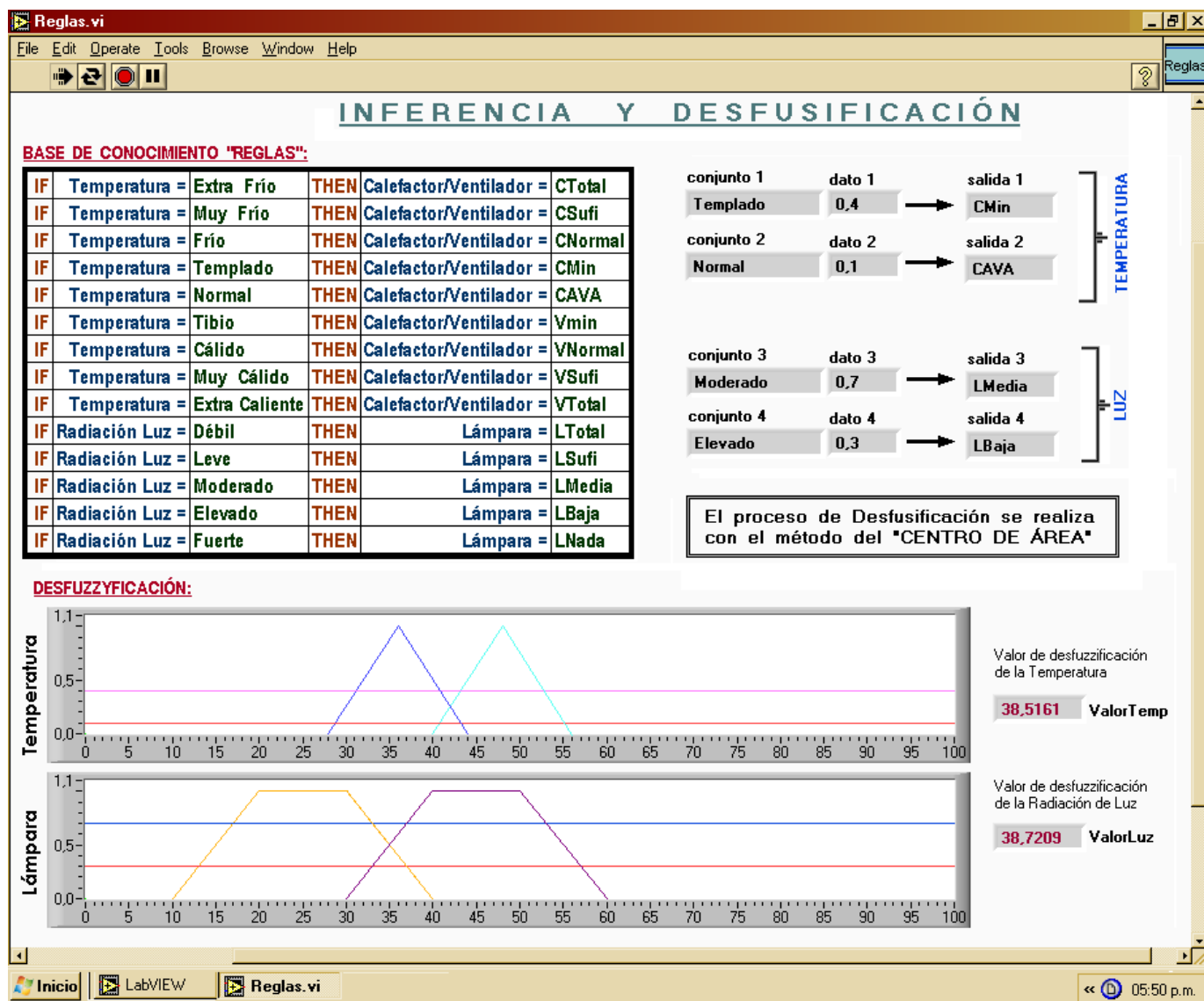


Figura 4.22 Interfaz LabView. Pantalla Reglas.

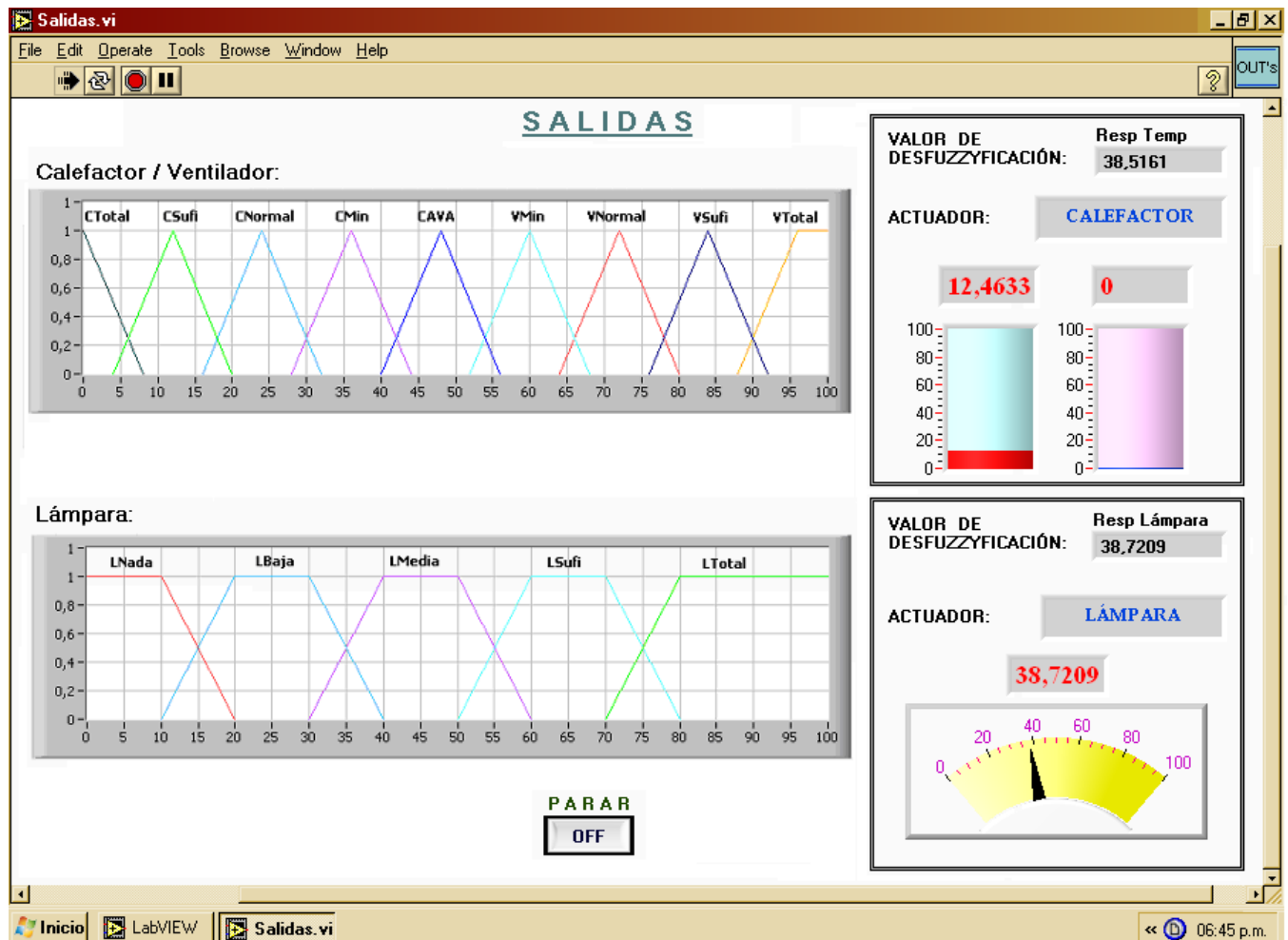


Figura 4.23 Interfaz LabView. Pantalla Salidas.

A continuación se presenta la interfaz de LabView del segundo programa, en donde no se programa lógica difusa, ya que dicho proceso debe ser programado en Step 7 para ser grabado en el PLC. No podemos hacer que los datos se resuelven en las dos partes porque darían respuestas erróneas.

En este segundo programa sólo hay comunicación serial con el PLC y conexión con Lookout.

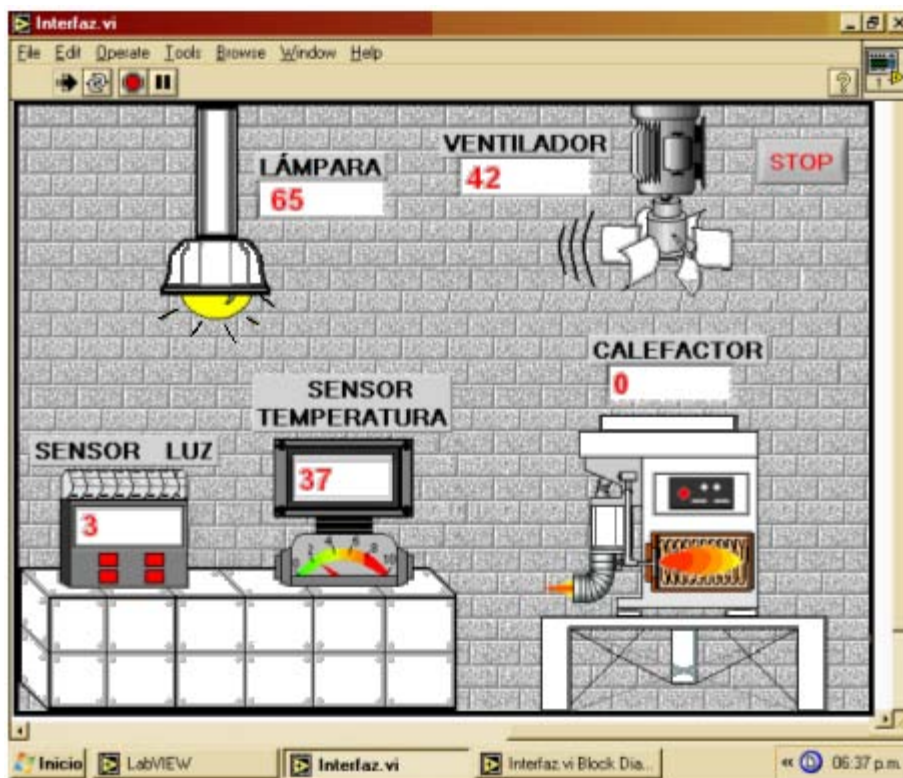


Figura 4.24 Interfaz de comunicación LabView – PLC.

4.3.3 DIAGRAMA DE BLOQUES DE LABVIEW.

La primera pantalla denominada entradas, debe realizar la comunicación serial para la lectura de sensores mediante los VIs: Serial Port Read y Bytes At Serial Port que efectúan la lectura del puerto serial COM 1, la recepción y transmisión serial se da en cadena de caracteres "string" por lo que se usa un convertidor de string a byte array para leer los dos bytes y para definir el primero y segundo dato se usa el index array.

También se debe mostrar los conjuntos difusos de temperatura y radiación de luz gráficamente y se logra mediante la introducción de los diferentes valores en un build array y éstos a su vez en un cluster bundle, para ser mostrados en un XY Graph.

Además se realiza el cálculo de las funciones de membresía de cada conjunto tanto para temperatura como radiación de luz, mediante la estructura Fórmula Node, en donde la señal que entregan los sensores pueden involucrar uno o dos conjuntos difusos. También se establecen las variables lingüísticas de cada conjunto, mediante una estructura Case (true / false).

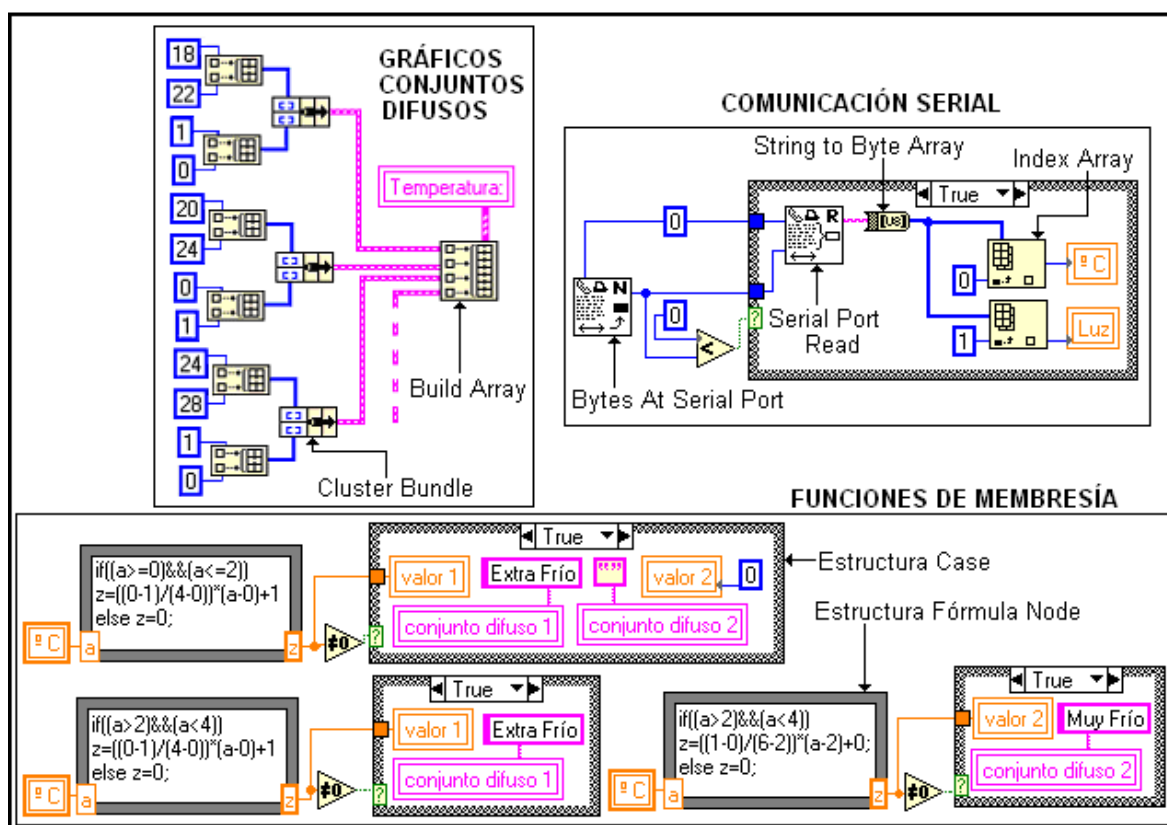


Figura 4.25 Parte principal de la pantalla entradas.

En la segunda pantalla denominada reglas, primero se llama al VI de las entradas con sus respectivos datos, luego se realizan las reglas que rigen la base de conocimientos tanto para la temperatura como para la radiación de luz.

También se muestra gráficamente las salidas difusas finales Calefactor/Ventilador y Lámpara con los valores de inferencia por el mínimo. Y por último se realiza el cálculo de los valores de desfusificación por el método del Centro de Área, utilizando una estructura Case numérico.

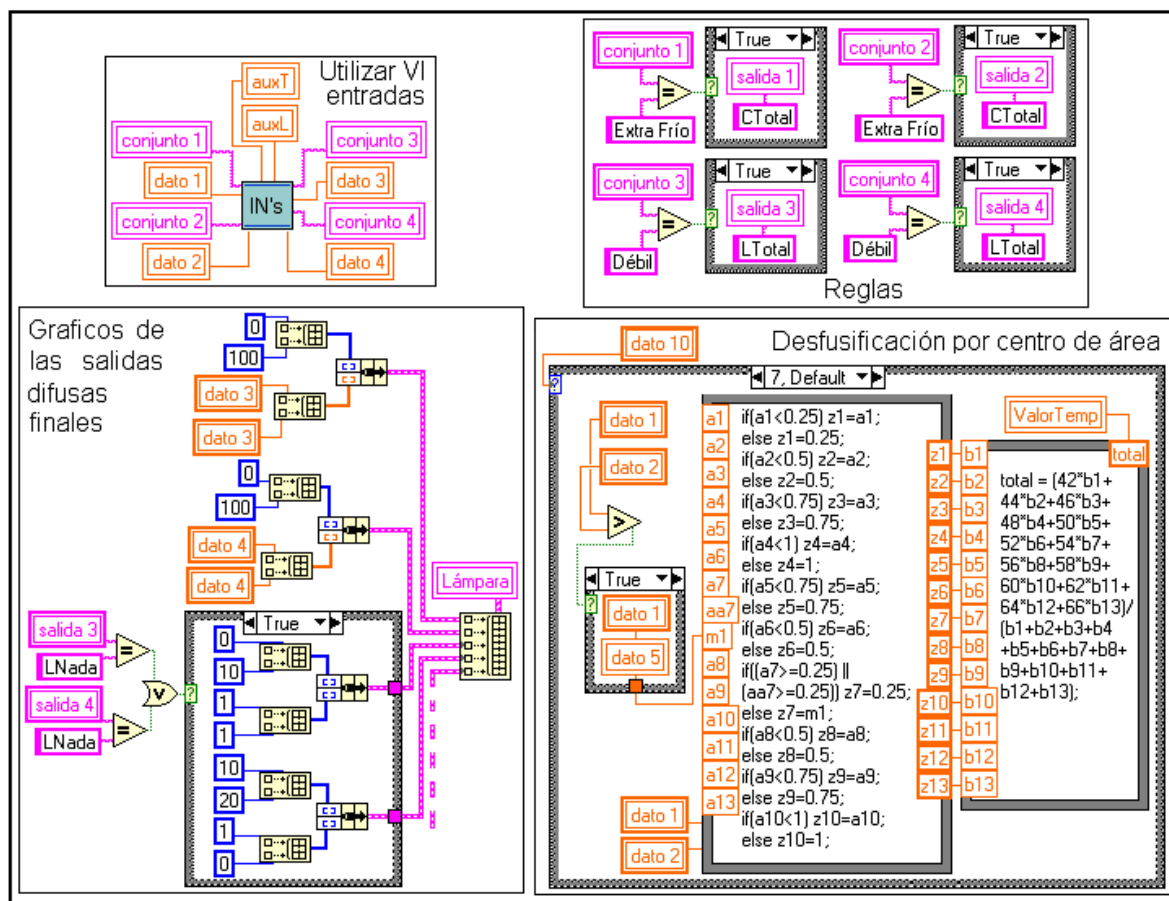



Figura 4.26 Parte principal de la pantalla reglas.

En la pantalla de salidas se utiliza el VI de reglas, se grafican las salidas de la misma forma que las entradas, se asignan los datos para el calefactor y ventilador a partir del valor de desfusicación, además se realiza la conexión con Excel, y la comunicación con el PLC a través del puerto serial COM 1, mediante el VI Serial Port Write se escriben los tres datos de: calefactor, ventilador y lámpara para los respectivos actuadores; se debe enviar un ENTER  dentro de la cadena enviada para que el PLC pueda finalizar la recepción del mensaje.

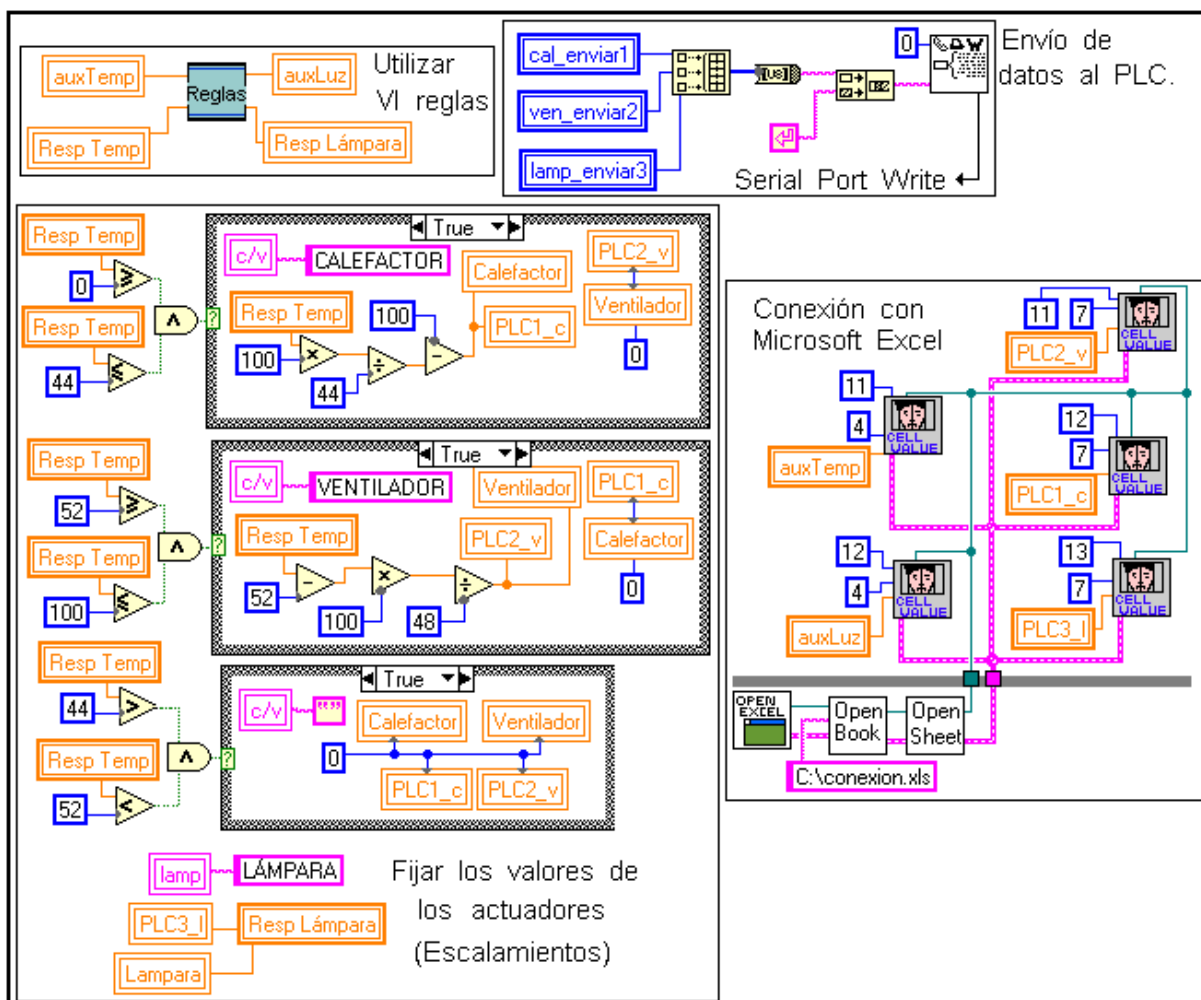


Figura 4.27 Parte principal de la pantalla salidas.

En la interfaz de comunicación LabView – PLC se realizan dos acciones, la primera corresponde a la escritura en una hoja de cálculo de Microsoft Excel para la respectiva comunicación con el programa en Lookout, la segunda es la lectura de todos los datos desde el PLC mediante las librerías: Bytes at serial port.VI y Serial port read.VI.

Para escribir en Excel, primero se debe abrir el programa mediante: Open excel and make visible, luego se abre el libro mediante: Open specific workbook.VI por consiguiente se abre la hoja de cálculo mediante el VI: Open specific workseet así se procede a escribir con: Write excel.VI, agregando la fila, columna y dato a ser escrito.

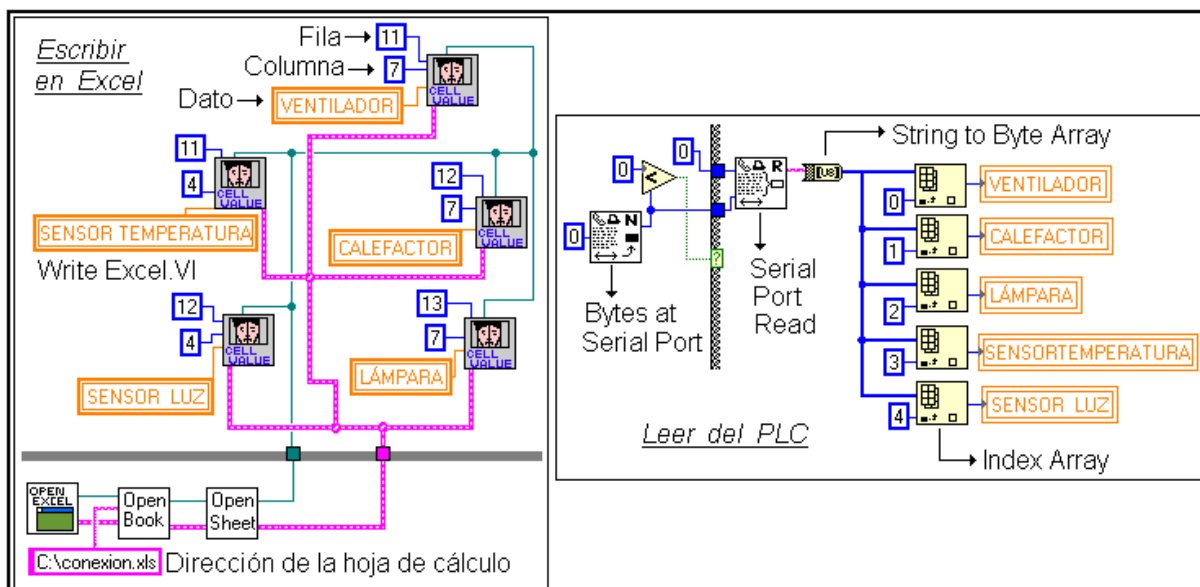


Figura 4.28 Parte principal de la Interfaz LabView – PLC.

Las operaciones de abrir Excel, libro y hoja de cálculo se deben realizar fuera de una estructura While, para que sean llamadas una sola vez y no genere error. La operación Bytes at serial port se debe realizar fuera de una estructura Case (Verdadero o Falso) para que los datos queden memorizados hasta que cambien de valor por lo que se comunica serialmente, caso contrario cuando no se recibe información el valor del dato es cero.

4.4 STEP 7

Step7 es un lenguaje de programación de propósito industrial a diferencia de Basic, C, C++, Java, etc. La versión del software que se utiliza en el presente proyecto es: Step7-Micro/Win 32 versión 4.0.

Su principal característica es que posee 3 tipos de programación:

- **KOP**: Es un esquema de contactos que se posicionan como en la realidad Fase y Neutro, tiene elementos de mando y control. Su juego de operaciones son símbolos en forma Ladder (escalera) para ejecutar los segmentos llamados Network desde arriba hacia abajo y de izquierda a derecha. Esta programación facilita el entendimiento y substancialmente el manejo de dicho lenguaje para el diseñador y programador industrial de aplicaciones tipo Scada.

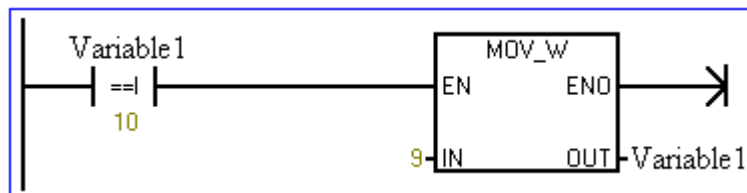


Figura 4.29 Ejemplo de programación KOP.

- **AWL:** Este tipo de programación se llama también Lista de instrucciones y permite crear programas de control introduciendo la nemotécnica de las operaciones, se adecua especialmente para los programadores expertos ya familiarizados con los sistemas de automatización y PLCs. El editor AWL también permite crear ciertos programas que no se pueden programar con los editores KOP ni FUP, debido a que AWL es el lenguaje nativo de la CPU, a diferencia de los editores gráficos en los que son aplicables ciertas restricciones para poder dibujar los diagramas correctamente⁷⁵. Un ejemplo de AWL:

NETWORK 1

LD	I0.0	//Load (Cargar) la entrada I0.0
LD	I0.1	//Load (Cargar) la entrada I0.1
LD	I2.0	//Load (Cargar) la entrada I2.0
A	I2.1	//Sumar I2.0 con I2.1
OLD		//(I2.0 + I2.1) ó I0.1
ALD		//I0.0 + [(I2.0 + I2.1) ó I0.1]
=	Q5.0	//Si se cumplen condiciones energizar Q5.0

- **FUP:** El editor llamado también Diagrama de funciones permite visualizar las operaciones en forma de cuadros lógicos similares a los circuitos de puertas lógicas, aquí no existen contactos ni bobinas como en el editor KOP, pero sí hay operaciones equivalentes que se representan en forma de cuadros. La lógica del programa se deriva de las conexiones entre esas operaciones de cuadro. Ello significa que la salida de una operación (por ejemplo, un cuadro AND) se puede utilizar para habilitar otra operación (por ejemplo, un temporizador) con objeto de crear la lógica de control necesaria.

⁷⁵ Ayuda de Step7-Micro/Win 32 versión 4.0. Manual de Usuario.

Estas conexiones permiten solucionar fácilmente numerosos problemas lógicos, al igual que con los otros editores.

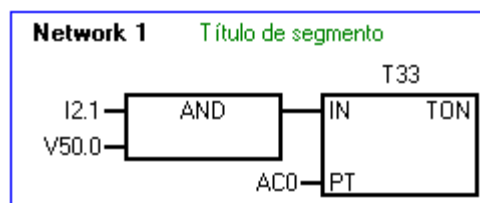


Figura 4.30 Tipo de programación FUP.

Step7 se compone de varias barras y de algunas ventanas:

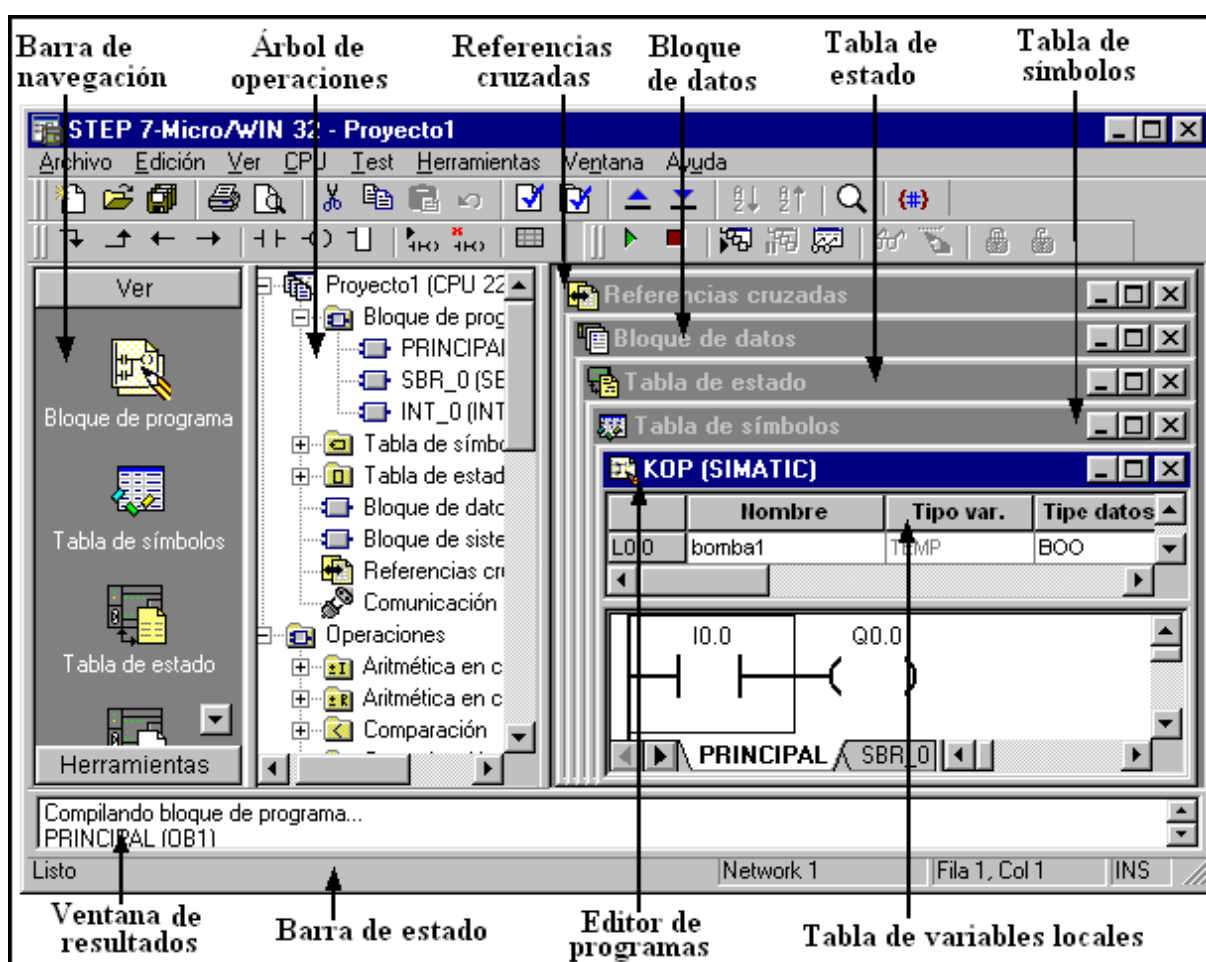


Figura 4.31 Componentes del entorno Step7-Micro/Win 32.

Barra de navegación.- Incorpora grupos de botones para facilitar la programación:

- Ver: se visualizan los botones Bloque de programa, Tabla de símbolos, Tabla de estado, Bloque de datos, Bloque de sistema, Referencias cruzadas y Comunicación.
- Herramientas: para visualizar los botones del asistente de operaciones, del asistente del TD 200, del asistente de control de posición, del Panel de control EM 253, así como del asistente de módems.

Árbol de operaciones.- Ofrece una vista en árbol de todos los objetos del proyecto y de todas las operaciones disponibles en el editor de programas actual (KOP, FUP o AWL⁷⁶). Este árbol puede ser editado y personalizado según las necesidades del usuario. Las operaciones utilizadas con frecuencia se pueden arrastrar y soltar en la carpeta "Favoritos".

Referencias cruzadas.- Permite visualizar las referencias cruzadas y los elementos utilizados en el programa.

Bloque de datos.- Sirve para visualizar y editar el contenido del bloque de datos.

Tabla de estado.- Permite observar el estado de las entradas, salidas y variables del programa. Es posible crear varias tablas de estado para visualizar elementos de diferentes partes del programa. Cada una de esas tablas tiene su propia ficha en la ventana "Tabla de estado".

Tabla de símbolos.- Sirve para asignar y editar símbolos globales (valores simbólicos que se pueden usar en cualquier unidad de organización del programa). Es posible crear varias tablas de símbolos. Al proyecto se puede agregar una tabla predefinida de símbolos de sistema S7-200.

Ventana de resultados.- Visualiza mensajes de información cuando se compila el programa de usuario o una librería de operaciones. Si se indican errores en esa ventana, haciendo doble clic en el mensaje de error, el segmento en cuestión se visualizará entonces en la ventana del editor de programas.

Barra de estado.- Informa acerca del estado de las funciones que se ejecutan en STEP 7-Micro/WIN.

⁷⁶ KOP: Esquema de contactos.
 AWL: Lista de instrucciones.
 FUP: Diagrama de funciones por cuadros lógicos.

Editor de programas.- Contiene la tabla de variables locales y la vista del programa correspondiente al editor (KOP, FUP, o AWL) utilizado en el proyecto actual. En la parte inferior se visualizan las unidades de organización del programa como: Principal, Subrutinas, Interrupciones, etc.

Tabla de variables locales.- Contiene asignaciones hechas a las variables locales (variables utilizadas por las subrutinas y las rutinas de interrupción). Las variables creadas en la tabla de variables locales utilizan la memoria temporal. El sistema se encarga de gestionar la asignación de direcciones. Las variables locales sólo se pueden utilizar en la UOP⁷⁷ donde se hayan creado.

Barra de menús.- Sirve para ejecutar funciones como: Archivo, Edición, Ver, CPU, Test, Herramientas, Ventana, Ayuda. Esta barra se encuentra en casi todos los programas de Windows. El menú Herramientas se puede personalizar.

Barras de herramientas.- Permiten acceder fácilmente a las funciones de STEP 7-Micro/WIN utilizadas con frecuencia. El contenido y el aspecto de cada una de las barras de herramientas se pueden personalizar.

El proyecto de tesis se creó en el tipo de programación KOP, por lo que se estudiará el juego de operaciones de KOP y no de AWL ni FUP.

4.4.1 CONCEPTOS BÁSICOS KOP.

El esquema de contactos es un lenguaje de programación gráfico con componentes similares a los elementos de un esquema de circuitos. Al programar se crean y se disponen componentes gráficos que conforman un segmento de operaciones lógicas. Para crear programas se dispone de los siguientes elementos:

- **Contactos:** Representan un interruptor por el que la corriente puede circular. La corriente circula por un contacto normalmente abierto sólo cuando el contacto está cerrado (es decir, cuando su valor lógico es "1"). De forma similar, la corriente circula por un contacto normalmente cerrado o negado (NOT) sólo cuando el contacto está abierto (es decir, cuando su valor lógico es "0").

⁷⁷ UOP: Unidad de Organización de Programa.

- **Bobinas:** Representan un relé, motor, actuador digital o una salida excitada por la corriente.
- **Cuadros:** Representan una función (por ejemplo, un temporizador, un contador o una operación aritmética) que se ejecuta cuando la corriente llega al cuadro.

Un segmento comprende esos elementos, representando un circuito completo. La corriente circula desde la barra de alimentación izquierda (representada en el editor KOP mediante una línea vertical en el lado izquierdo de la ventana) a través de los contactos cerrados para excitar las bobinas o los cuadros.

Algunas reglas para comenzar la programación:

- Cada segmento debe comenzar con un contacto.
- Un segmento no se puede terminar con un contacto.
- Un segmento no puede comenzar con una bobina.
- Las bobinas se utilizan para terminar un segmento.
- Un solo segmento puede contener varias bobinas, en tanto que las bobinas se encuentren en ramificaciones paralelas de ese segmento en particular. No se puede disponer más de una bobina en serie (es decir, en una sola línea horizontal de un segmento no puede haber varias bobinas).
- Si un cuadro tiene una salida de habilitación ENO, la circulación de corriente continúa detrás del cuadro, eso significa que después del cuadro se pueden disponer más operaciones.
- En un mismo circuito de un segmento se pueden conectar en serie varios cuadros que dispongan de salidas ENO.
- Si un cuadro no tiene ENO no se puede colocar ninguna otra operación después del mismo.

Un ejemplo de posicionamiento de contactos, bobinas y cuadros a continuación:

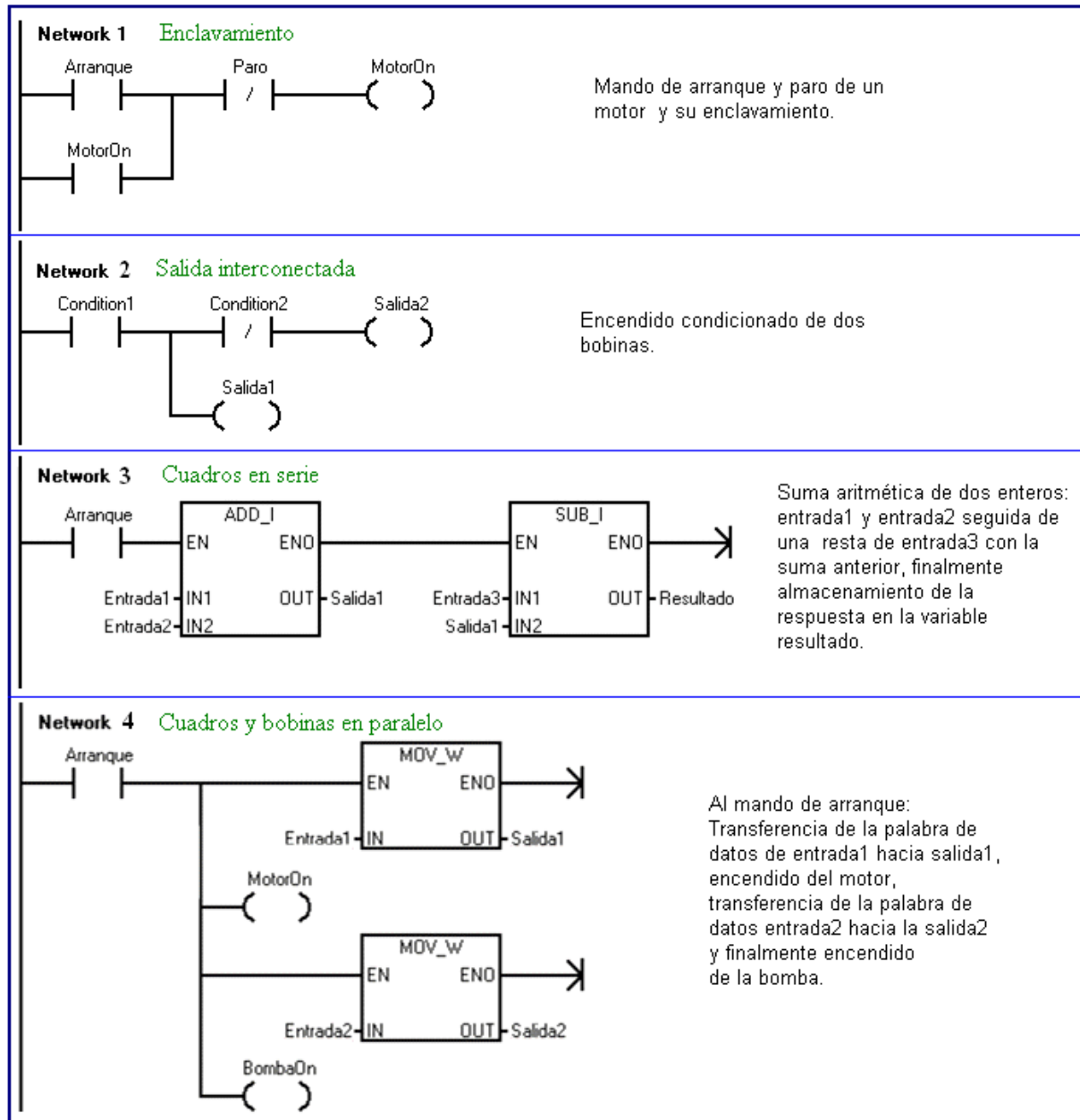


Figura 4.32 Ejemplos de posicionamiento de contactos, bobinas y cuadros.

Direccionamiento:

I para denominar entradas, **Q** para denominar salidas. Junto con el distintivo de entrada o salida aparece el parámetro 0.4, 1.2 ó 4.7. El parámetro consiste en una combinación:

0, 1, 4 → byte. 4, 2, 7 → bit.

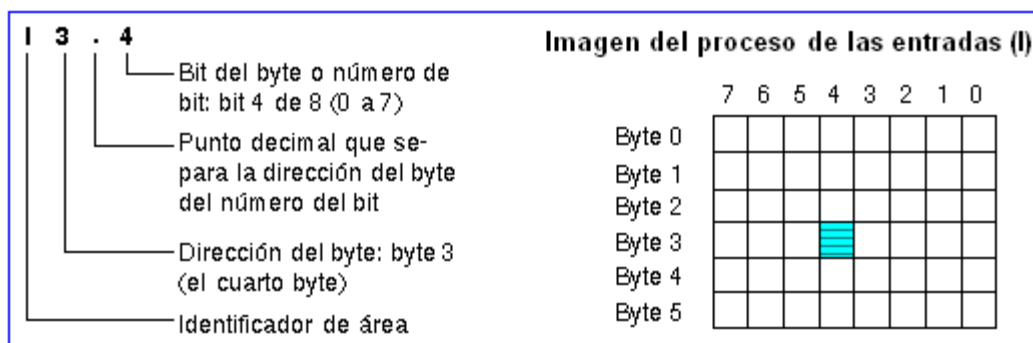


Figura 4.33 Estructura de un operando y direccionamiento.

En caso de marcas se llama con la letra M, en marcas espaciales con SM, en variables la letra V seguida de la letra de memoria, por ejemplo en formato byte con: VB, en palabra: VW, en doble palabra: VD. El formato doble palabra consta de 4 bytes ó 32 bits. Sirve para trabajar con decimales o números reales o con enteros de 32 bits, a diferencia de la palabra que trabaja con enteros de 16 bits. Ejemplos: VB0, VB1, VB2, VB3, VW6, VW8, VW10, VD14, VD18, VD22, VD26, etc. Los bytes se llaman de uno en uno, las palabras se llaman de dos en dos, las palabras dobles se llaman de cuatro en cuatro; esto se hace para que no exista solapamiento en la memoria.

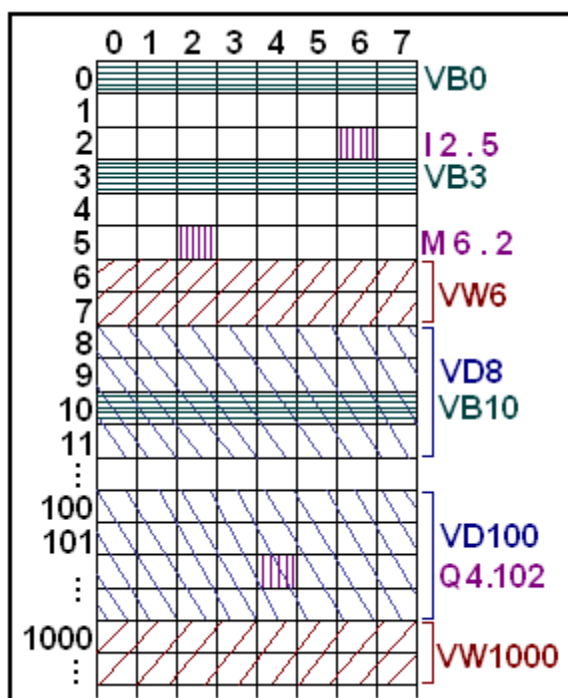


Figura 4.34 Asignación de memoria.

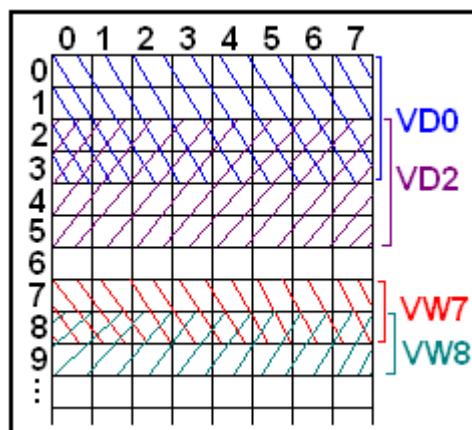


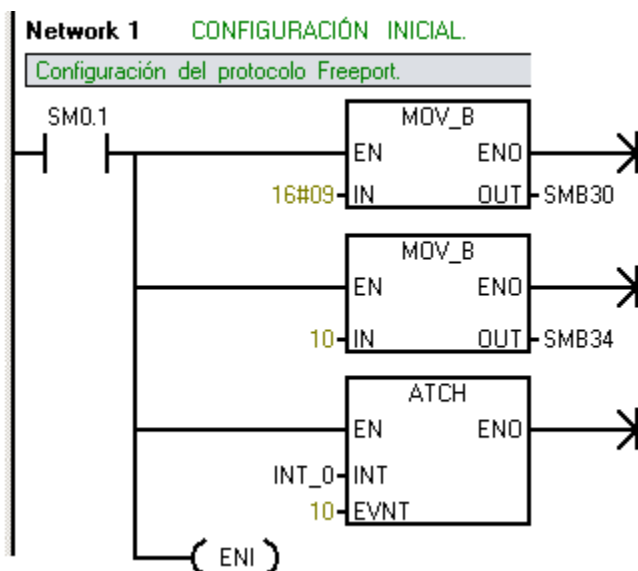
Figura 4.35 Solapamiento de variables.

4.4.2 PROGRAMA KOP.

Se presenta el programa KOP del proyecto de tesis realizando todo el proceso de lógica difusa.

PRINCIPAL:

Entrada 1: Sensor de Temperatura (AIW0)	de 0 a 50 grados Centígrados.
Entrada 2: Sensor de Luz (AIW2)	de 0 a 10 niveles de luz.
Salida 1: Actuador Calefactor	de 0 a 100% de su acción.
Salida 2: Actuador Ventiladores	de 0 a 100% de su acción.
Salida 3: Actuador: Lámpara.	de 0 a 100% de su acción.



La configuración del protocolo Freeport se realiza formando un byte de los datos que se muestran en la figura 4.36 y estableciéndolos en la marca especial SMB30 para el puerto cero (COM1) o SMB130 para el puerto uno (COM2):

Dirección		Formato de bit	Formato de bit							
Puerto 0	Puerto 1		MSB				LSB			
			7							0
			p	p	d	b	b	b	m	m
SMB30		pp:								
	SMB130									
SM30.6 y SM30.7	SM130.6 y SM130.7		0	0	= Sin paridad					
			0	1	= Paridad par					
		1	0	= Sin paridad						
		1	1	= Paridad impar						
SM30.0 y SM30.1	SM130.0 y SM130.1	d:		0	= 8 bits de datos por carácter					
				1	= 7 bits de datos por carácter					
SM30.2 a SM30.4	SM30.2 a SM30.4	bbb:		0	0	0	= 38.400 bit/s			
				0	0	1	= 19.200 bit/s			
				0	1	0	= 9.600 bit/s			
				0	1	1	= 4.800 bit/s			
				1	0	0	= 2.400 bit/s			
				1	0	1	= 1.200 bit/s			
				1	1	0	= 115.200 bit/s			
				1	1	1	= 57.600 bit/s			
SM30.0		mm:		0	0	= Protocolo de interface punto a punto (PPI/modo esclavo)				
	SM130.0			0	1	= Protocolo Freeport				
SM30.1	SM130.1			1	0	= PPI/modo maestro				
				1	1	= Reservado (cambia al ajuste estándar PPI/modo esclavo)				

Figura 4.36 Byte para la configuración de SMB30 o SMB130.

En Step7 se debe anteceder al número hexadecimal por el formato: 16#

p	p	d	b	b	b	m	m
0	0	0	0	1	0	0	1

= 16#09

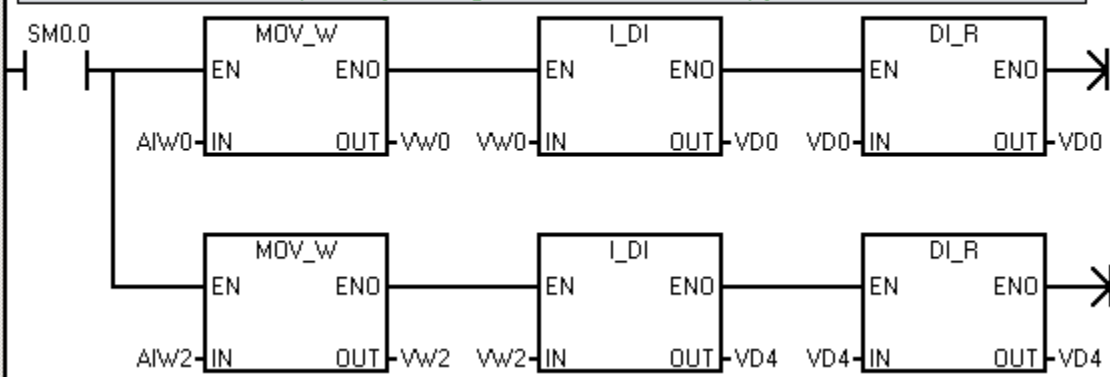
Por tanto se está configurando los parámetros: Sin paridad, con 8 bits de datos por carácter, a una velocidad de transmisión de 9600 bits/s.

Se transfiere también a la marca SMB34 un valor de 10 milisegundos para una transmisión temporizada. Y se asocia la interrupción 0 con el evento 10 para disparar la transmisión. ENI se utiliza para habilitar la interrupción.

Se realiza la lectura de los sensores AIW0 para el sensor de temperatura y AIW2 para el sensor de luz y se transforman los valores a números reales:

Network 2 LECTURA DE SENSORES

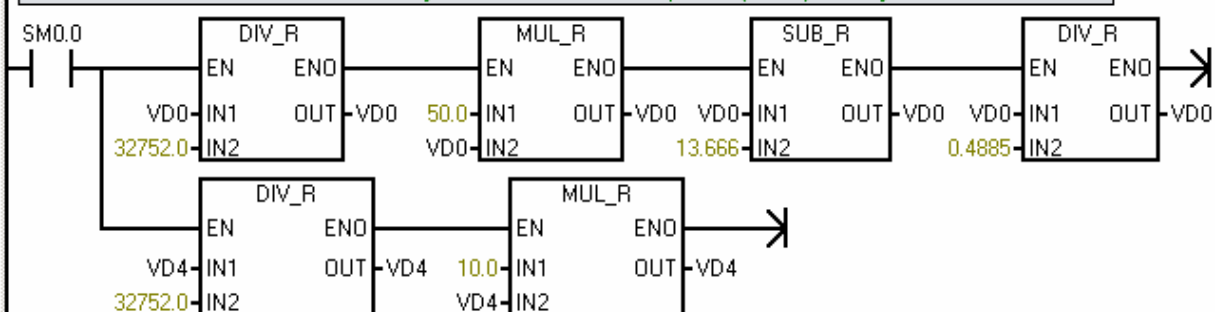
Lectura de los sensores Temperatura y Luz, asignación a las variables: Vtemp y Vluz; conversión a número real.



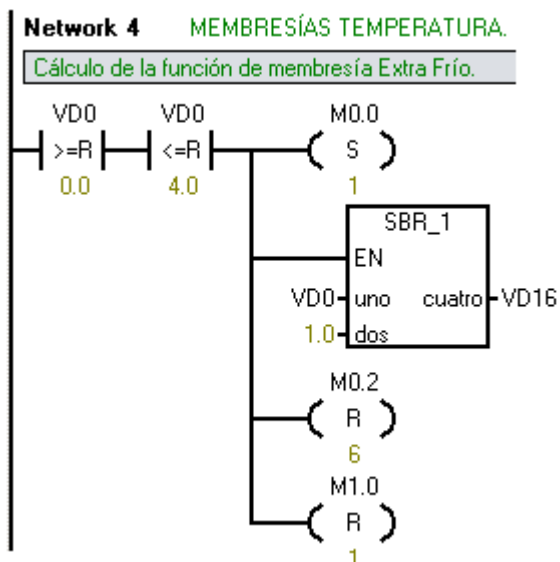
Luego se efectúa el escalamiento de los sensores debido a que el módulo analógico EM235 entrega una lectura en formato palabra de 0 a 32752. Para el sensor de temperatura se realiza un cálculo matemático para conseguir resultados más cercanos a la realidad, el termistor es NTC (coeficiente de temperatura negativo), la explicación de la fórmula está detallada en el capítulo 3 de Hardware, en el subcapítulo Sensores, en la figura 3.3, esta fórmula representa la pendiente de una recta.

Network 3 ESCALAMIENTO.

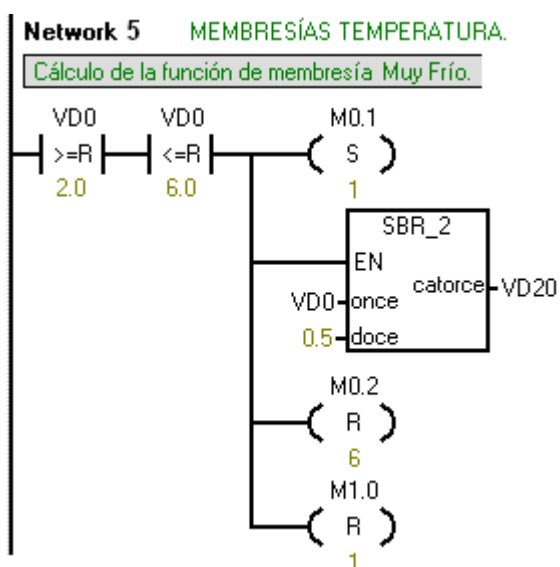
Transformación de medidas de los sensores y escalamiento del formato palabra (32752) a 50°C y nivel 10 de luminosidad.



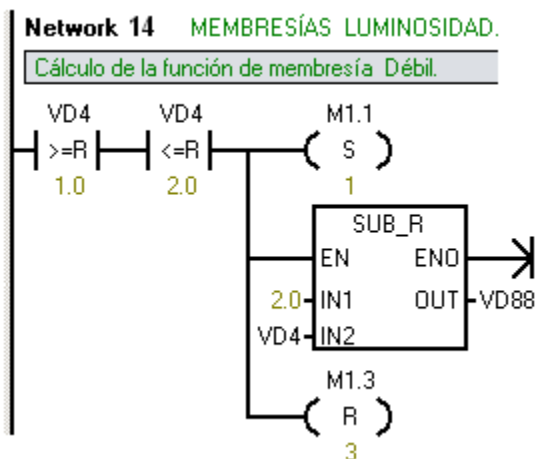
Se calcula las funciones de membresía para cada conjunto difuso de temperatura por medio de la subrutina 1, y se asignan las marcas para la futura Base de Conocimientos:



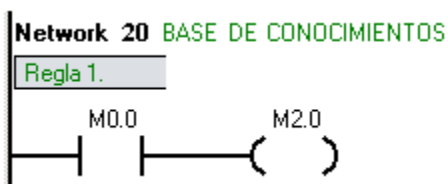
Se calculan todas las funciones de membresía para cada conjunto difuso de temperatura por medio de la subrutina 2:



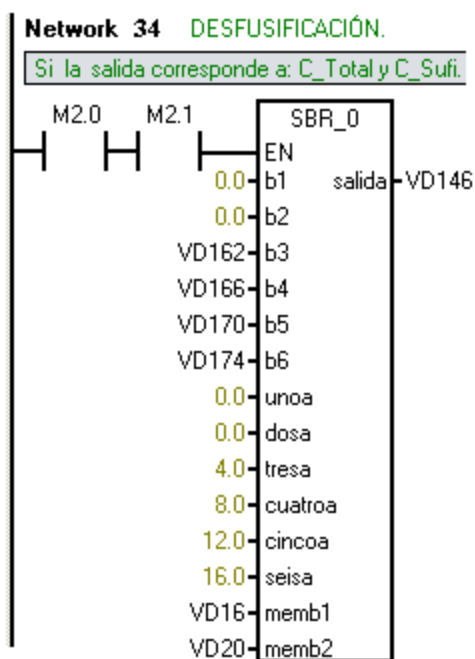
Se calculan todas las funciones de membresía para cada conjunto difuso de la luminosidad por medio de la fórmula “pendiente de la recta”:

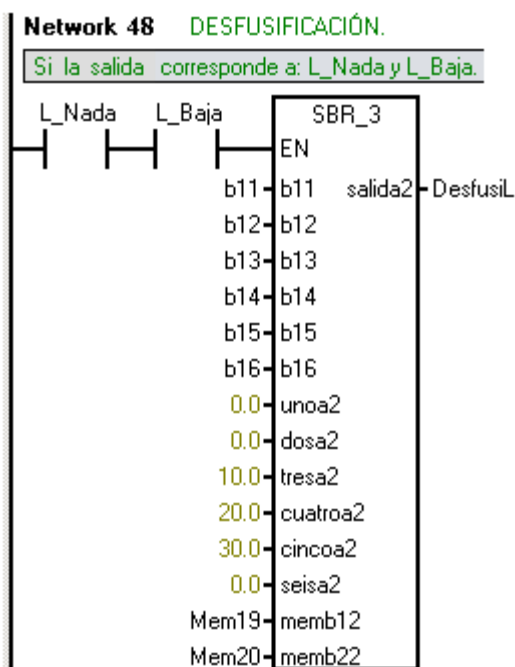


Luego se establece cada una de las reglas de la Base de Conocimientos mediante las marcas asignadas anteriormente:



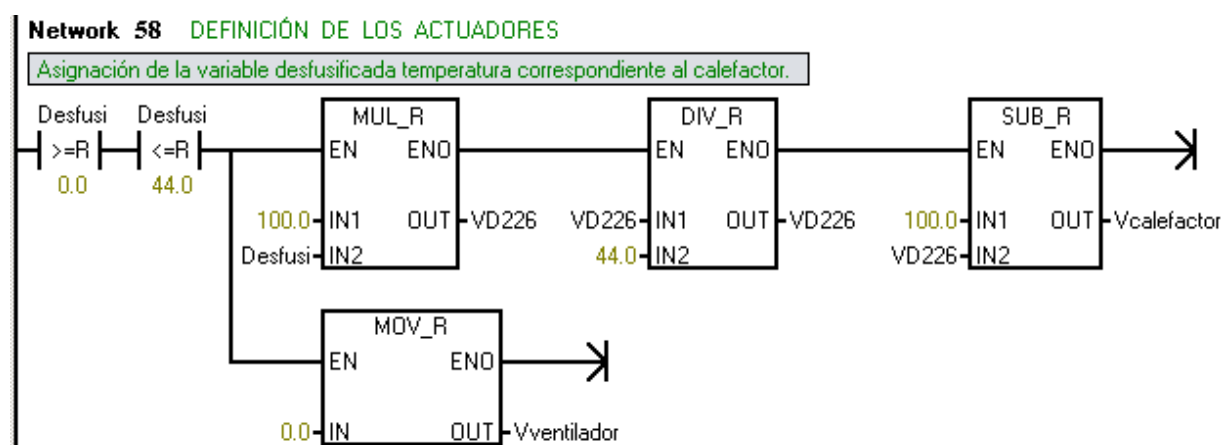
Realizamos la desfusificación para encontrar la salida correspondiente a la temperatura por medio de la fórmula “Centro de Área”, a través de la subrutina 0:





Efectuamos la desfusificación para encontrar la salida correspondiente a la luz por medio de la fórmula “Centro de Área”, a través de la subrutina 3.

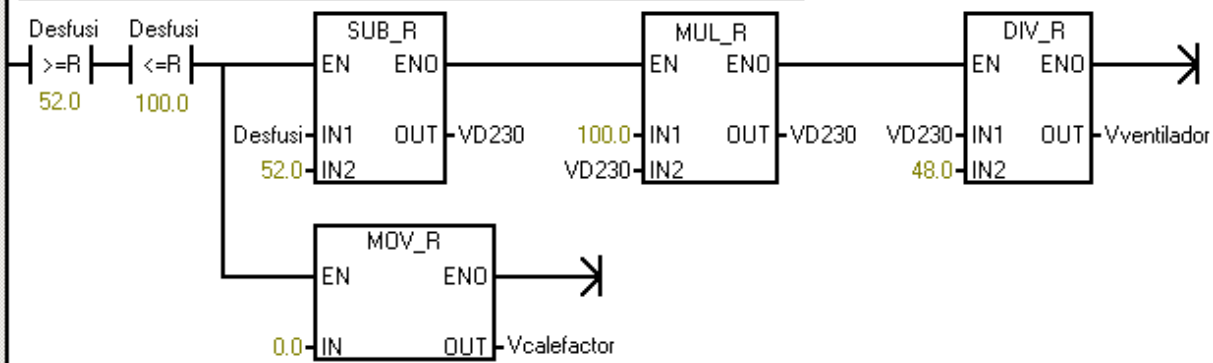
Se definen los valores de la salida difusa de temperatura que corresponde al actuador Calefactor y se realiza el cálculo escalar para mostrar los valores en porcentajes:



Se definen los valores de la salida difusa de temperatura que corresponde al actuador Ventilador y se realiza el cálculo para mostrar los valores en porcentajes:

Network 59 DEFINICIÓN DE LOS ACTUADORES

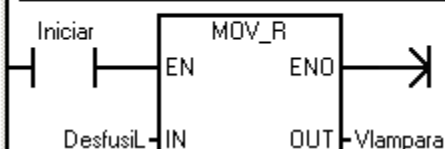
Asignación de la variable desfusificada temperatura correspondiente al ventilador.



Se define el valor de la salida difusa de luz correspondiente al actuador Lámpara:

Network 60 DEFINICIÓN DE ACTUADORES

Variable desfusificada correspondientes a la lámpara.

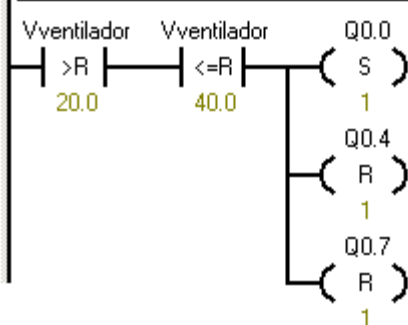


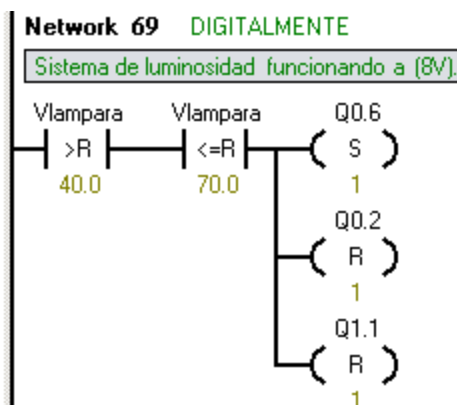
Debido a que el módulo analógico EM235, sólo posee una salida analógica, y el proyecto requiere de 3 salidas analógicas, se estableció que la Calefacción se utilizará normalmente con la salida analógica que nos brinda el módulo, y las 2 que faltan que son Ventilador y Lámpara, se simulan con 2 bits, es decir 4 estados: el primero apagado, el segundo a 5 voltios, el tercero a 8 voltios y el cuarto a 12 voltios.

Digitalmente asignamos los voltajes al Ventilador y a la Lámpara:

Network 64 DIGITALMENTE

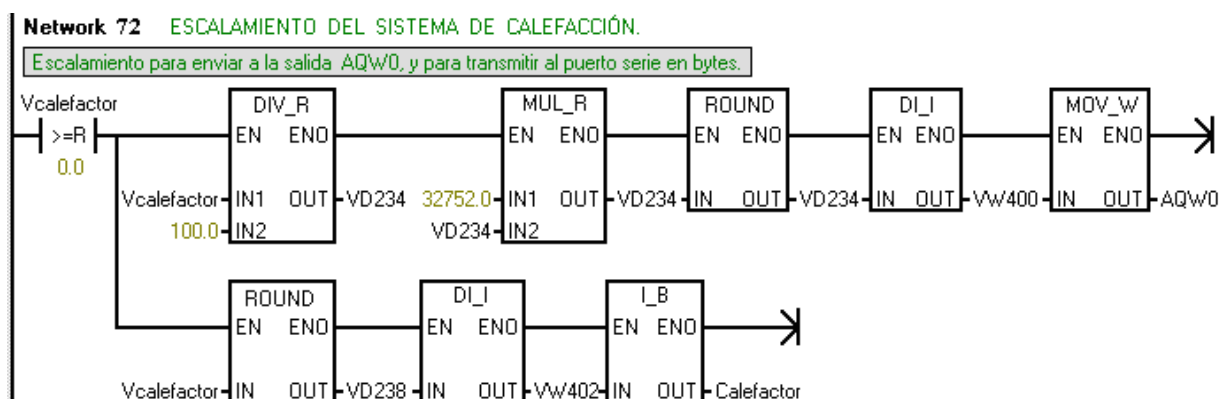
Sistema de ventilación funcionando al mínimo (5V).



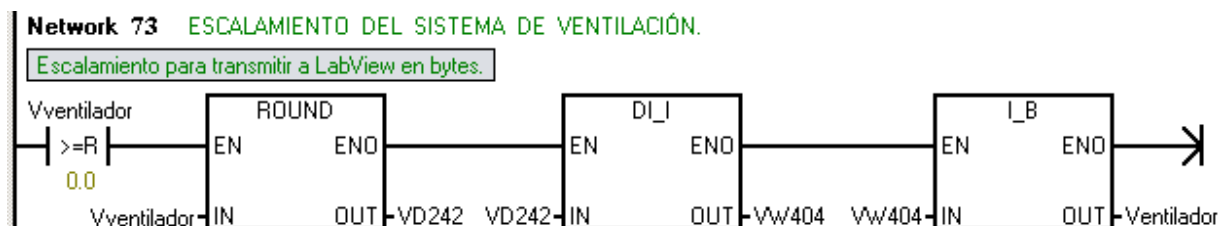


Escalamiento de Calefacción para asignar a AQW0 en formato palabra de 32752 y asignar al módulo EM235.

Además se realiza un escalamiento para transmitir en bytes:

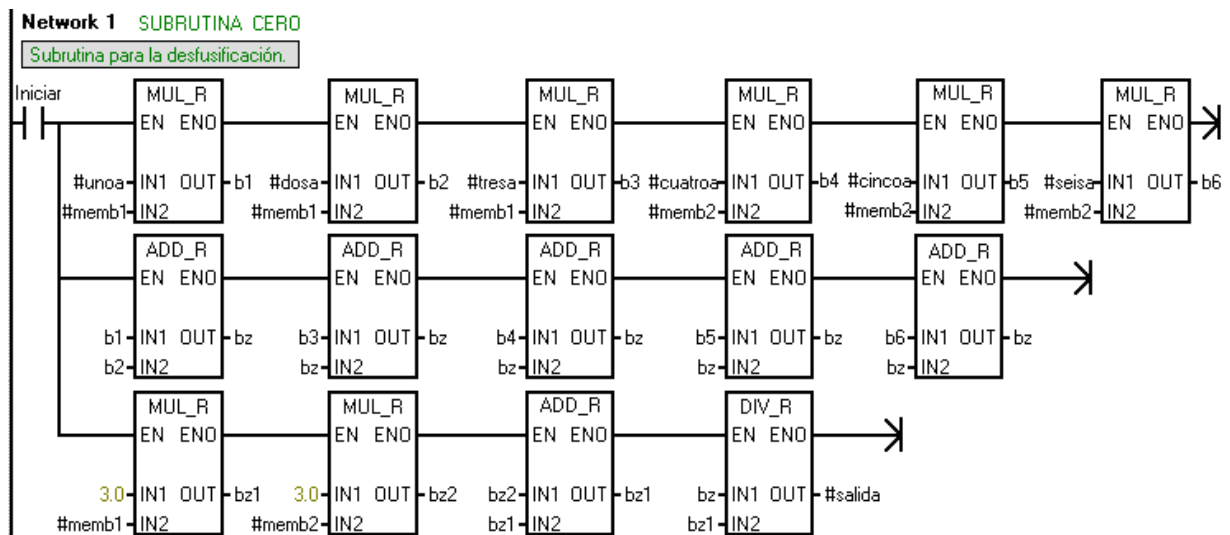


Al igual que el calefactor se debe efectuar el escalamiento al Ventilador y a la Lámpara para poder transmitir por el puerto serial en bytes, también se debe realizar el escalado para el sensor de temperatura y de luz.



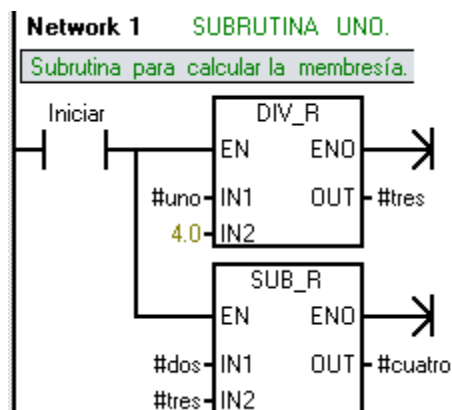
SUBROUTINA CERO:

Las variables que se encuentran anteceditas por el símbolo de # son temporales y sólo son auxiliares para el cálculo matemático. En esta subrutina se realiza el cálculo de la desfusificación por medio del método del Centro de Área, estudiado en el capítulo 1. Se define la salida difusa correspondiente a la entrada de temperatura.



SUBROUTINA UNO:

Esta subrutina realiza el cálculo matemático de las funciones de membresía de la entrada temperatura, mediante la fórmula de la pendiente estudiada en el capítulo 1, definiendo el proceso de fusificación:



SUBROUTINA DOS:

Se efectúa el cálculo matemático de las funciones de membresía de la entrada luz, mediante la fórmula de la pendiente, así se define el proceso de fusificación. Utiliza el mismo procedimiento que la subrutina 1 con la diferencia de los nombres de las variables.

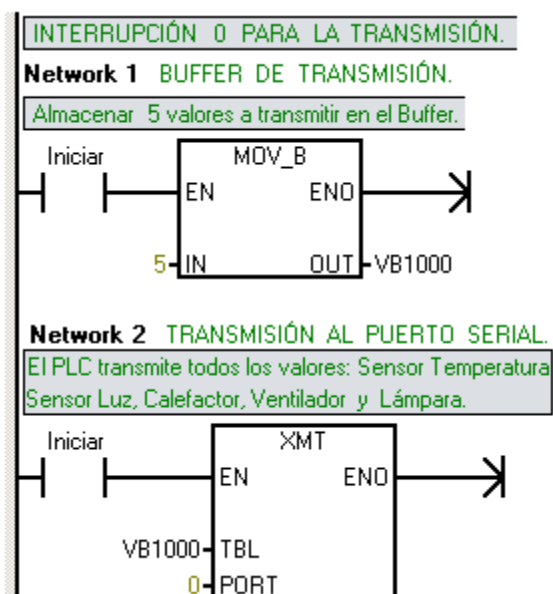
SUBROUTINA TRES:

En esta subrutina se elabora el cálculo de la defusificación por medio del método del Centro de Área. Se define la salida difusa correspondiente a la entrada de luminosidad, utiliza el mismo procedimiento que la subrutina 0 con la diferencia de los nombres de las variables.

INTERRUPCIÓN CERO:

En la interrupción cero se realiza la transmisión del PLC hacia el puerto serial mediante un Buffer de tipo byte (VB1000), se transmiten 5 valores: sensor temperatura, sensor luz, calefacción, ventilador, lámpara.

Estos valores deben estar almacenados desde VB1001 en adelante, ya que se transmiten en forma de tabla de bytes, a través de la operación XMT por el puerto 0 (COM1):



FIN del primer programa.

A continuación se detalla el segundo programa en donde se realiza la comunicación bidireccional y NO se realiza lógica difusa ya que este proceso se debe efectuar en LabView.

En el primer ciclo del PLC se configuran las marcas especiales:

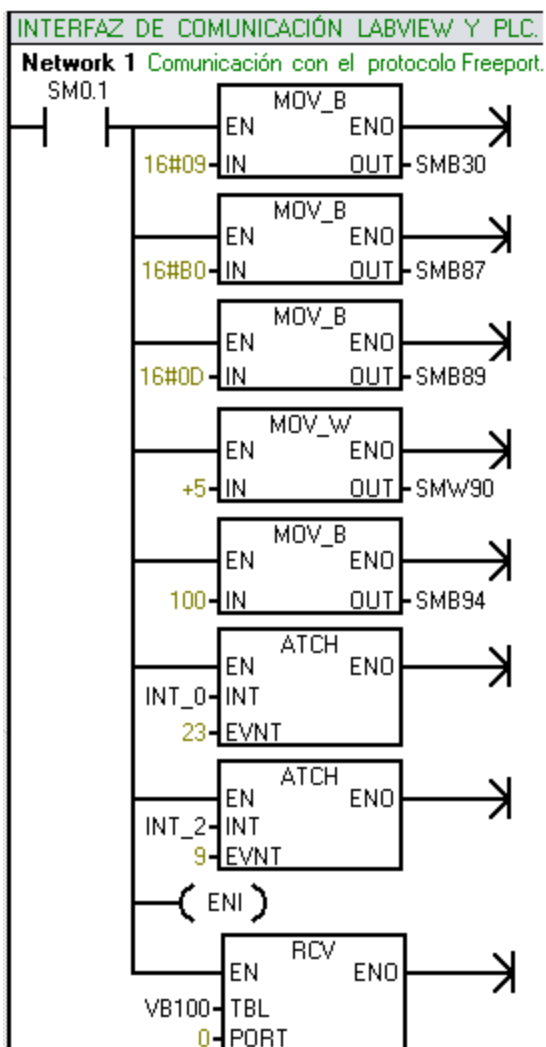
- SMB30.- Protocolo Freeport: por el puerto COM1, sin paridad, con 8 bits de datos por caracter, a una velocidad de 9600 bits por segundo.
- SMB87.- Para definir las condiciones de inicio y fin del mensaje.

	CERO	UNO
SM87.0	Siempre cero	Nunca 1
SM87.1	Ignorar condición BREAK	Utilizar condición BREAK al inicio de mensajes
SM87.2	Ignorar SMW92	Finalizar recepción si excede el tiempo en SMW92
SM87.3	Temporizador entre caracteres	Temporizador de mensajes
SM87.4	Ignorar SMW90	Usar SMW90 para detectar condición de inactividad
SM87.5	Ignorar SMB89	Usar SMB89 para detectar fin del mensaje
SM87.6	Ignorar SMB88	Usar SMB88 para detectar comienzo del mensaje
SM87.7	Recepción de mensajes inhibida	Recepción de mensajes habilitada

Figura 4.37 Configuración de la marca SMB87.

Con las condiciones: Ignorar Break, Ignorar SMW92, temporizador entre caracteres, considerando SMW90 para la condición de inactividad, usando SMB89 para detectar el fin del mensaje, ignorando SMB88 y para habilitar la recepción de mensajes; se tiene el número: 10110000 en binario, 176 en decimal, y B0 en hexadecimal.

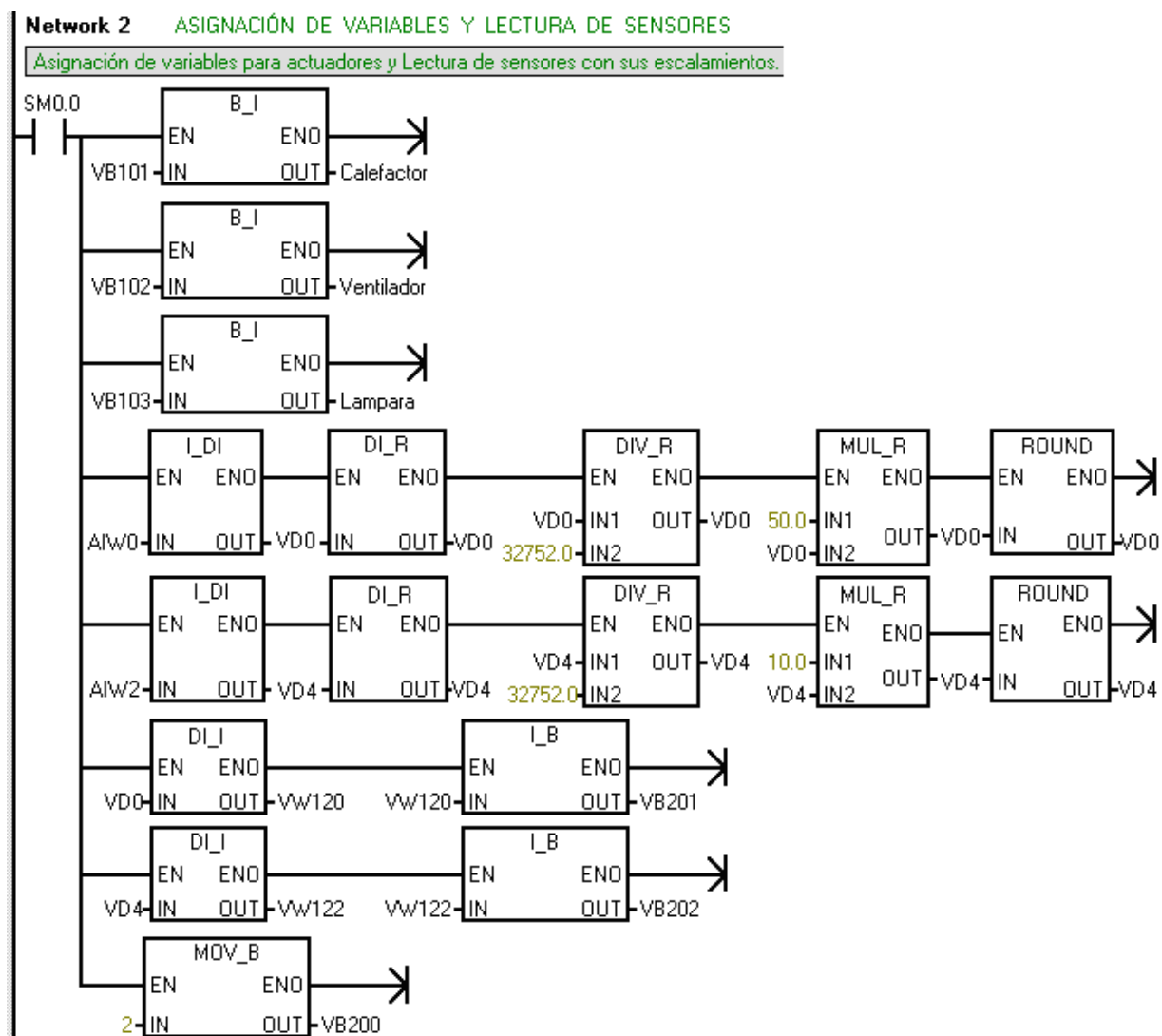
- SMB89.- Ajustar el caracter final recibido al valor hexadecimal 0D "cambio de línea".
- SMW90.- Ajustar el tiempo de línea inactiva a 5 milisegundos "timeout".
- SMB94.- Definir el número máximo de caracteres a 100.

PRINCIPAL:

Se prosigue con la asociación de la interrupción 0 con el evento 23 que corresponde a recepción de mensajes finalizada. Se asocia la interrupción 2 con el evento 9 que concierne a la transmisión de mensaje finalizada. Seguido se habilita las interrupciones y la recepción del mensaje a través del buffer tipo tabla de bytes VB100 por el puerto COM1 "0".

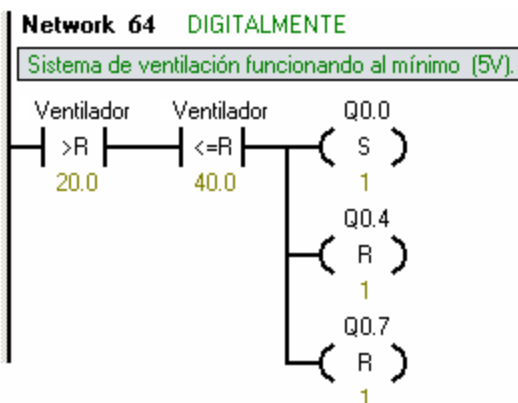
Luego se realiza la asignación de las variables para los actuadores: Calefactor, Ventilador, Lámpara como también la lectura de sensores, AIW0 para el sensor de temperatura y AIW2 para el sensor de luz escalando los valores con sus respectivas dimensiones ya que el módulo EM235 entrega en formato palabra de 0 a 32752.

A si mismo se realizan transformaciones a bytes para poder transmitirlos por el puerto serial en el futuro; asignamos al buffer VB200 el valor de 2 ya que vamos a enviar los valores de los 2 sensores hacia LabView en VB201 y en VB202.

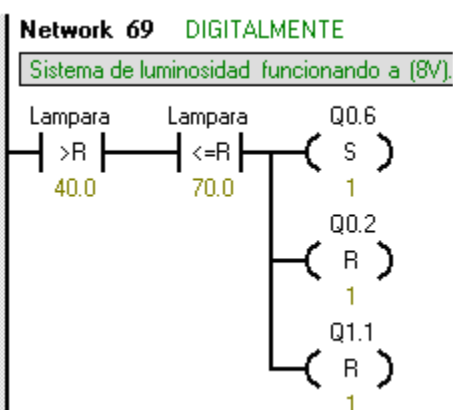


Debido a que el módulo analógico EM235, sólo posee 1 salida analógica, la Calefacción se utilizará normalmente con ella, y las 2 que faltan que son Ventilador y Lámpara, se simulan con 2 bits, es decir 4 estados: el primero apagado, el segundo a 5 voltios, el tercero a 8 voltios y el cuarto a 12 voltios.

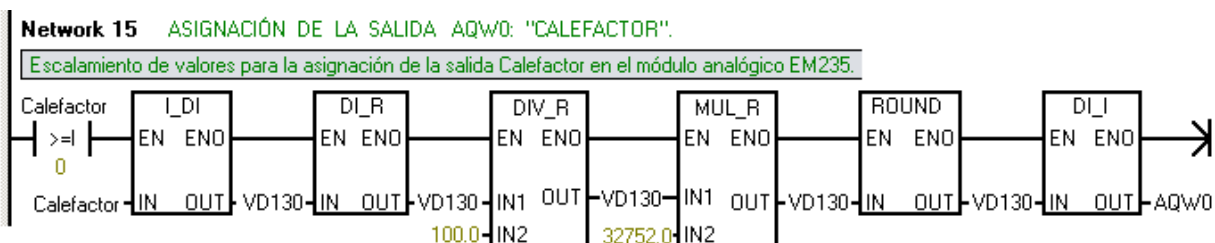
Digitalmente asignamos los voltajes al Ventilador:



Digitalmente asignamos los voltajes a la Lámpara:



Finalmente en el bloque principal se efectúa la asignación de la salida hacia el actuador Calefactor mediante el módulo analógico EM235 luego de escalar el valor al formato palabra de 0 a 32752:



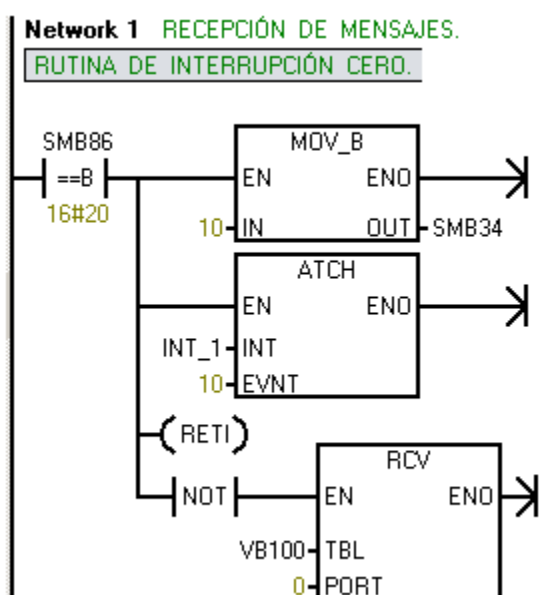
INTERRUPCIÓN CERO:

En esta rutina se realiza el control de recepción de mensajes, mediante la marca especial SMB86 para el puerto COM1 o puerto 0 (cero), a continuación se detalla su configuración:

Dirección SM		Formato de bit	Byte de estado de recepción de mensajes							
Puerto 0	Puerto 1		MSB				LSB			
			7							0
SMB86			n	r	e	0	0	t	c	P
	SMB186									
SM86.7		n:	1	= El usuario ha inhibido la recepción de mensajes.						
	SM186.7									
SM86.6		r:	1	= Ha finalizado la recepción de mensajes: error en los parámetros de entrada o falta condición inicial o final.						
	SM186.6									
SM86.5		e:	1	= Carácter final recibido						
	SM186.5									
SM86.2		t:					1	= Ha finalizado la recepción de mensajes: ha transcurrido la temporización.		
	SM186.2									
SM86.1		c:					1	= Ha finalizado la recepción de mensajes: se ha alcanzado el número máximo de caracteres.		
	SM186.1									
SM86.0		p:						1	= Ha finalizado la recepción de mensajes: error de paridad.	
	SM186.0									

Figura 4.38 Configuración de la marca SMB86.

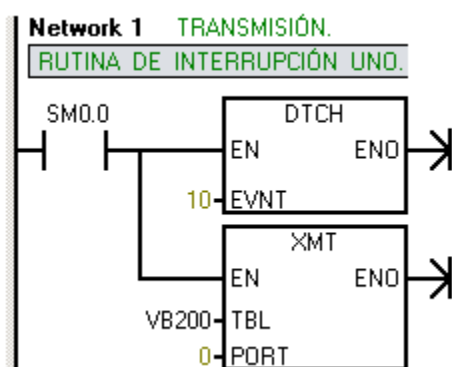
En nuestro caso, la recepción de mensajes finaliza por el carácter final recibido “ENTER”, que se enviará desde LabView. Por consiguiente el resultado es: 00100000 en binario, 32 en decimal y 16#20 en hexadecimal.



Se establece la marca SMB34 con intervalos de 10 milisegundos para la rutina de interrupción temporizada para la transmisión; se asocia la interrupción 1 con el evento 10 de interrupción temporizada, se crea un retorno condicionado de interrupción para que reinicie la recepción si se ha cortado por otro motivo o porque ya se transmitió.

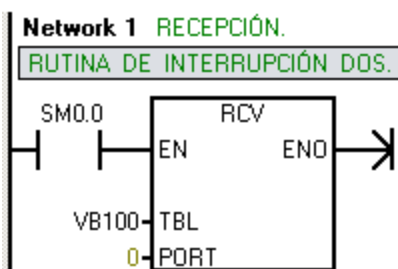
INTERRUPCIÓN UNO:

Se desasocia la interrupción temporizada para que no se enclave la transmisión y luego se envía el buffer VB200 tipo tabla de bytes que contiene los valores de los sensores, a través del comando XMT al puerto 0 (cero) o COM1.



INTERRUPCIÓN DOS:

Se realiza la recepción del mensaje a través del buffer tipo tabla de bytes que contiene el valor de los actuadores; por el puerto de comunicación 0 (cero) o COM1 mediante la operación RCV.



FIN del segundo programa.

CAPÍTULO 5. ANÁLISIS DEL SISTEMA.

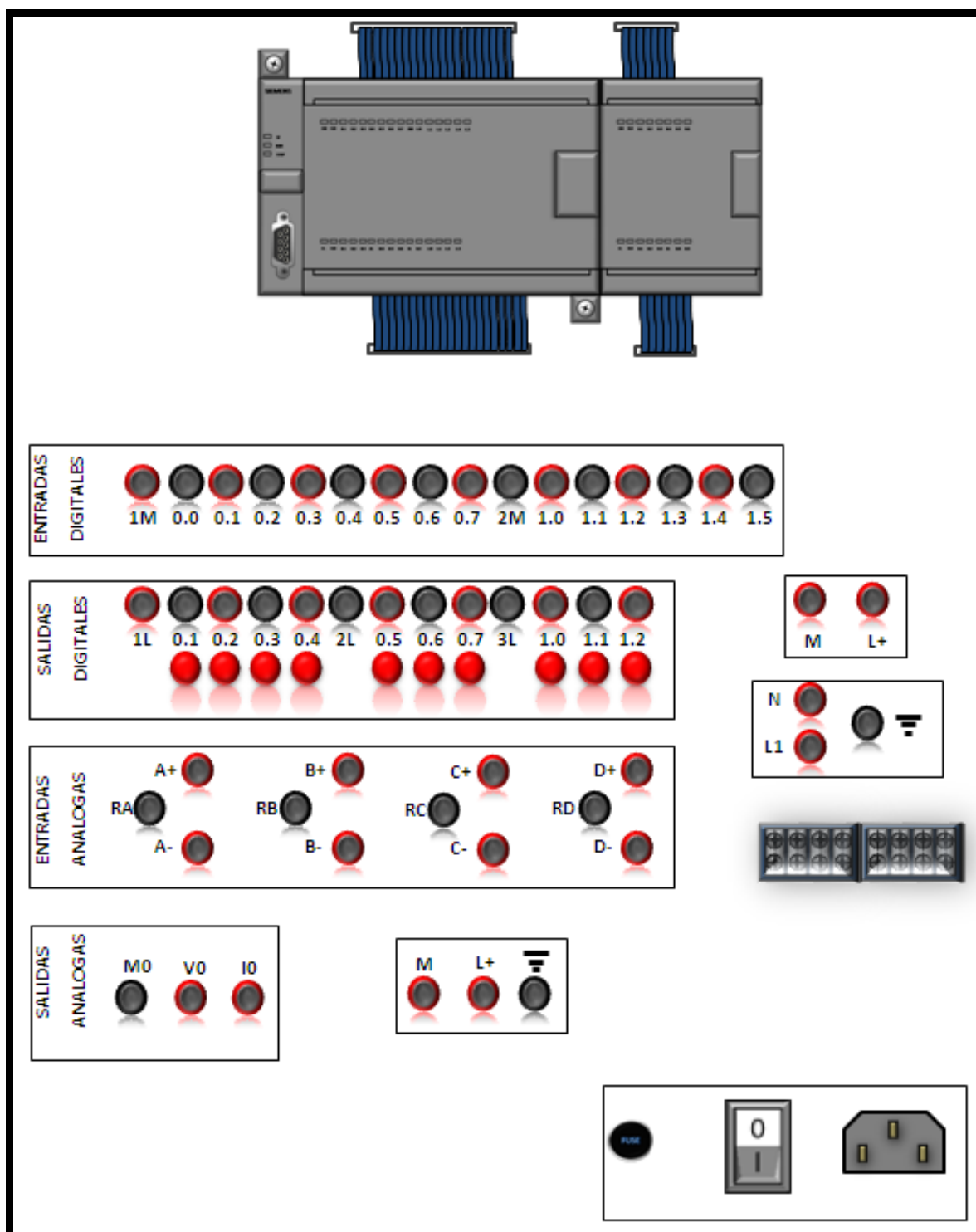
5.1. ANÁLISIS TÉCNICO ECONÓMICO.

Ítem	Cantidad	Descripción	Precio Unitario	Precio Total
1	1	PLC Siemens CPU224, alimentación 110/220 Vca, entradas 24Vdc salidas relé, memoria 12Kbyte 14DI/10DO, 1 puerto RS-485.	475,86	475,86
2	1	EM235 Módulo de 2 entradas analógicas, y 3 salidas análogas resolución 12 bits	345,05	345,05
3	1	Cable interface PPI para comunicación y programación S7-200/PC (USB).	203,94	203,94
4	1	Sensor de temperatura Termistor NTC de automóvil	3,50	3,50
5	1	Sensor de Luz Fotorresistencia	1,20	1,20
6	2	Ventiladores de 12v.	6,55	13,10
7	1	Calefactor (secador de cabello).	15,79	15,79
8	2	LCDs	8,20	16,40
9	2	Microcontroladores PICs 16F876A	7,50	15,00
10	5	Tarjetas electrónicas con circuito impreso. (c/u diferente valor)		40,00
11	1	Fuente de poder (5v, 8v, 12v). Vcc.	27,50	27,50
12		Elementos electrónicos como: borneras, potenciómetros, resistencias, cables, leds, conectores, switches, pulsadores, chips, transistores, circuitos integrados, triacs, bus de datos, terminales, indicadores, fusibles, tornillos, cables de poder, regletas, riel Din, otros.	60,20	60,20
13	1	Foco halógeno dicróico de 50 w.	6,10	6,10
14	1	Estructura del prototipo de prueba	50,00	50,00
TOTAL				1273,64

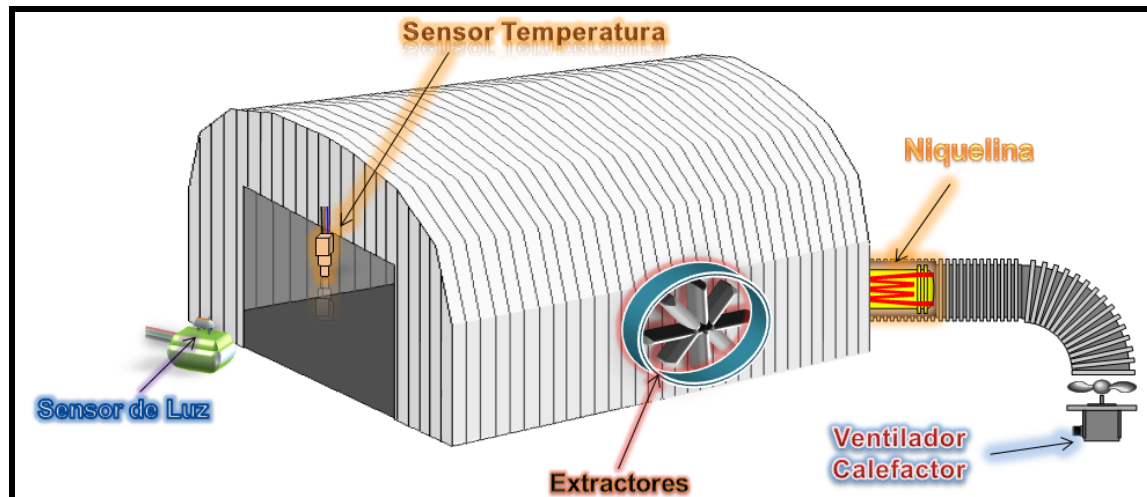
Figura 5.1 Tabla de gastos realizados.

5.2. MANUAL DE USUARIO.

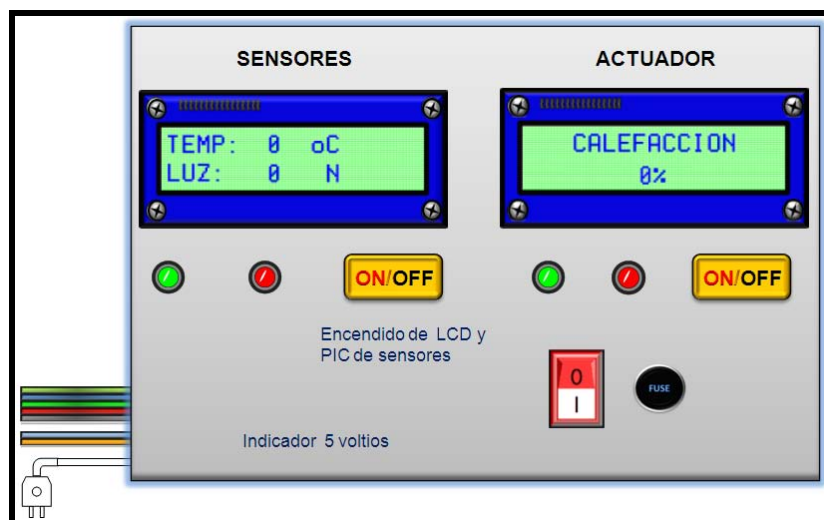
Partes del proyecto.



Tablero de conexiones de PLC y Modulo de Extensión EM235.



Hardware donde se encuentran montados los sensores y actuadores.



Visualizador de datos para Sensores y Actuador

REQUISITOS PARA HARDWARE.

Para el funcionamiento del proyecto se debe disponer de los siguientes componentes.

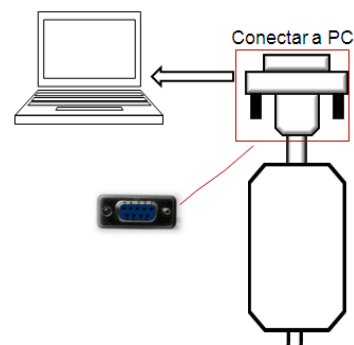
- Autómata Programable PLC (Siemens S200 CPU 224).
- Modulo de Extensión Analógico EM235.
- Computadora.
- Cable de comunicación PLC (Convertidor de Interfaces).
- Fuente de Poder de PC.
- Sensor de luz. (Fotorresistencia 4,7 KΩ a luz media).

- Sensor de Temperatura (Termistor NTC 560 Ω a 25° C).
- Circuitos eléctricos.

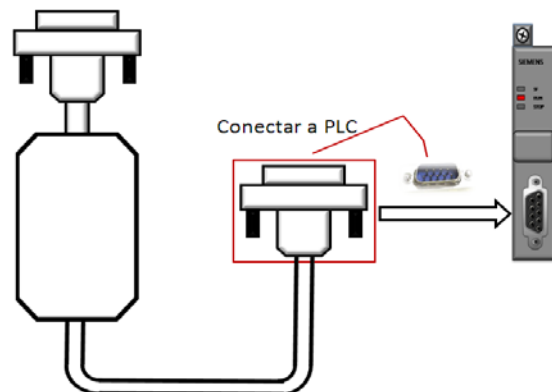
1. Conecte el cable de poder de 110 V_{AC}, al tablero donde se encuentra montado el PLC.



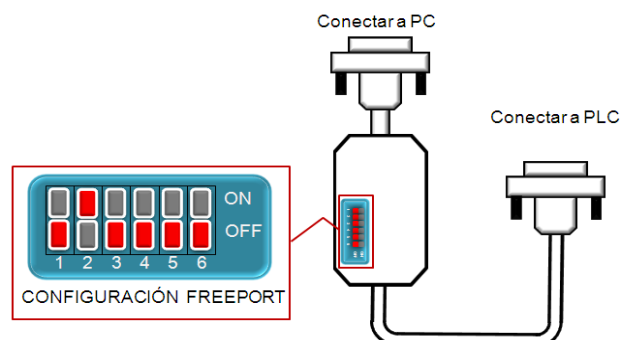
2. Conecte el cable de comunicación, al puerto serial (COM 1), (conector hembra DB9).



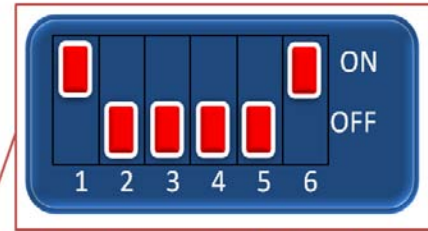
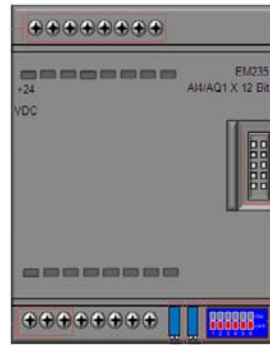
3. Conecte el cable de comunicación, al PLC (conector macho DB9).



4. Para la comunicación Freeport. Verifique que los dipswitch, del cable de comunicación se encuentren: el **2 ON** y los demás **1, 3, 4, 5 y 6 en OFF**.

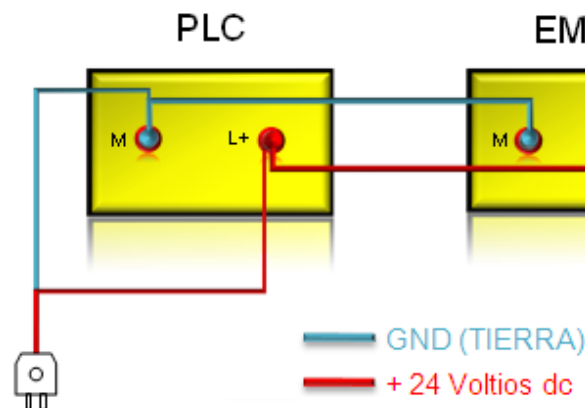


5. Verifique en el modulo de extensión **EM235**, los dipswitch se encuentren configurados el **1** y **6** **ON** y los demás **2, 3, 4** y **5** en **OFF**, para el funcionamiento de 0 a 5 voltios.



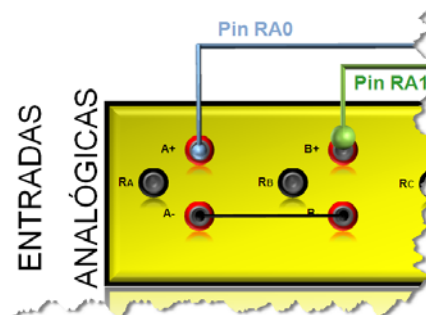
Configuración Entradas y Salidas Analógicas

6. Conecte los cables (conectores tipo banana) al modulo de extensión **EM235**, para que este funcione. De la siguiente manera:
- Del terminal **M** del PLC al terminal **M** del modulo **EM235**.
 - Del terminal **L+** del PLC al terminal **L+** del modulo **EM235**.

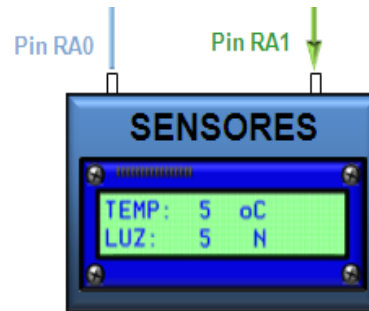


7. Conecte los cables (conectores tipo banana) de los sensores, al modulo de extensión EM235.

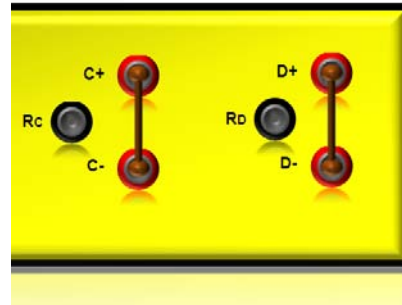
- El sensor de temperatura al terminal **A+** del PLC y al visualizador de datos de los **SENSORES**, Pin **RA0** del microcontrolador 16F876A.
- El sensor de Luz al terminal **B+** del PLC y al visualizador de datos de los **SENSORES**, Pin **RA1** del microcontrolador 16F876A.



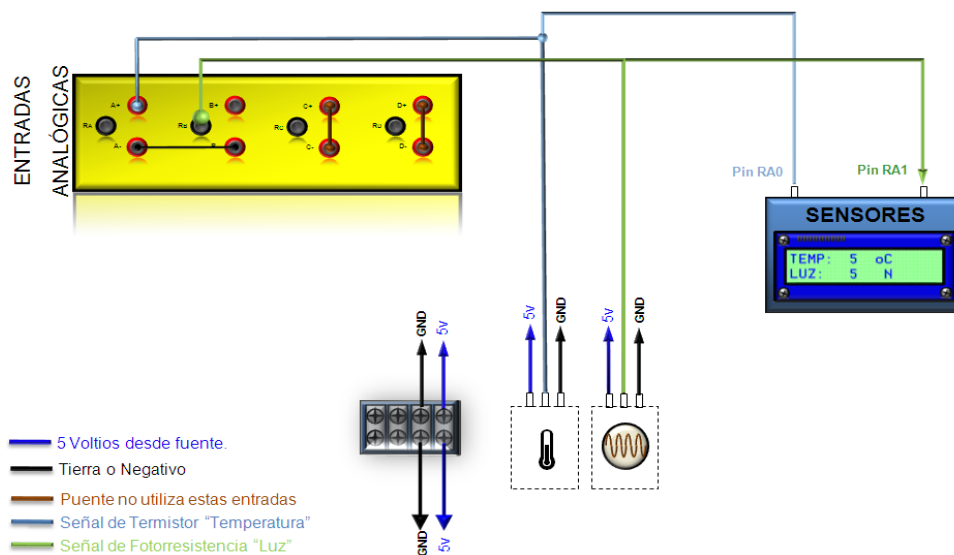
- c) Los terminales **A-** y **B-** debe conectarlos a **negativo** de la fuente.



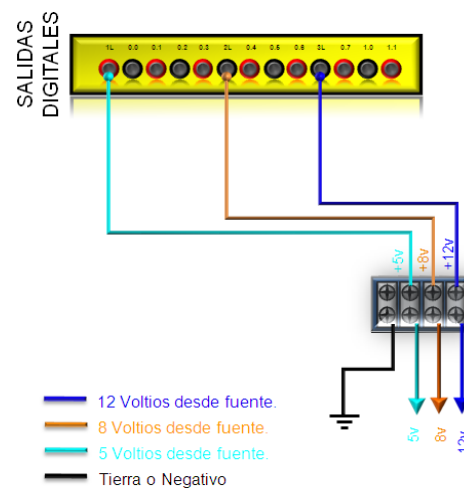
- d) Puentear los siguientes terminales entre si:
- **C+** y **C-**, debido a que no se utiliza esta entrada analógica.
 - **D+** y **D-**, debido a que no se utiliza esta entrada analógica.



8. Las conexiones de los sensores quedarían como se muestra en el grafico.

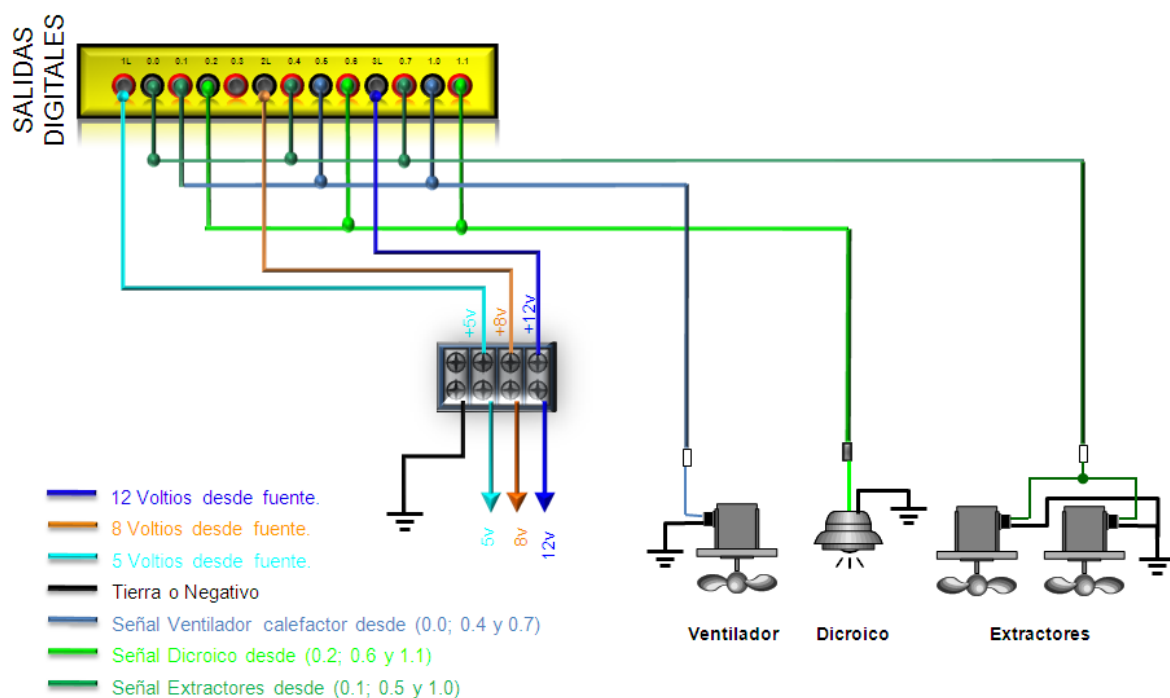


9. Conecte la alimentación de las fuentes, para el funcionamiento de los actuadores: extractores, ventilador del calefactor, lámpara (dicróico), de la

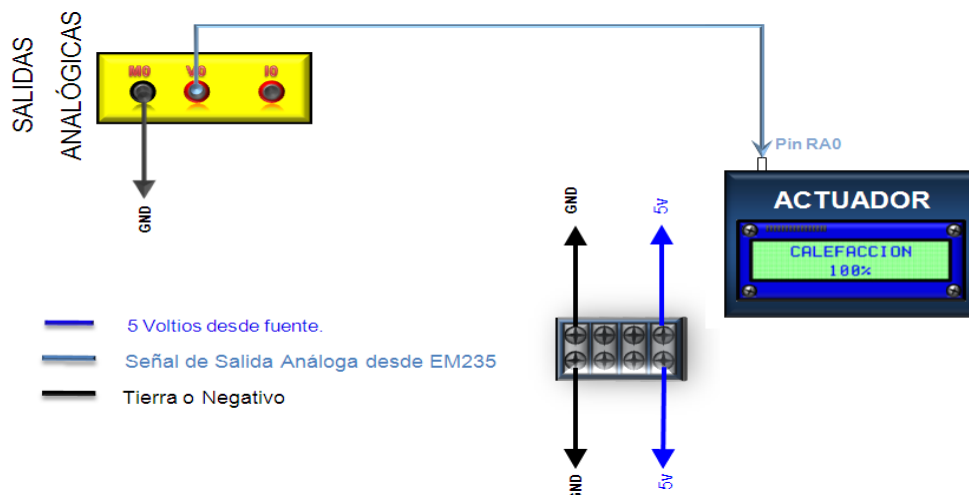


siguiente manera:

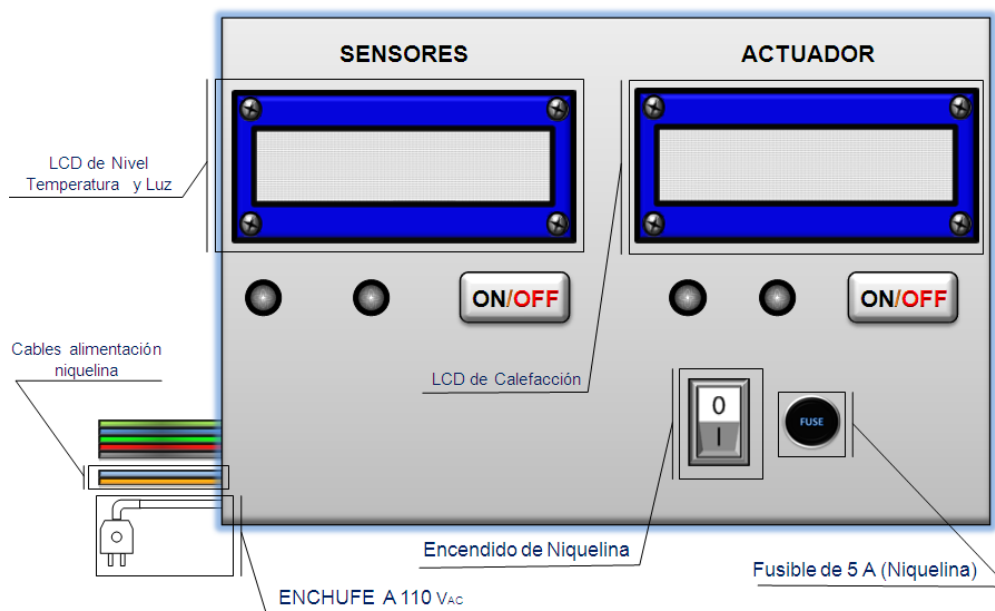
- 1L para 5 voltios.
 - 2L para 8 voltios.
 - 3L para 12 voltios.
10. Conecte los cables desde las salidas digitales hacia los actuadores, de la siguiente manera:
- a) Para extractores a las salidas digitales 0.0 ; 0.5 ; 0.7
 - b) Para ventilador del calefactor a las salidas digitales 0.1 ; 0.4 ; 1.0
 - c) Para la lámpara (dicróico) a las salidas digitales 0.2 ; 0.6 ; 1.1



11. Conecte los cables:
- a) Desde la salida analógica **V0** al visualizador de datos **ACTUADOR**, Pin **RA0** del microcontrolador 16F870.
 - b) Desde de las salidas analógicas **M0** a tierra de la fuente.



12. Para el funcionamiento de la niquelina, conecte el cable **ENCHUFE A 110 V_{AC}**, que se encuentra en el visualizador de datos de los Sensores y Actuador. Este será controlado por el circuito de los Actuadores dependiendo el valor de temperatura que se reciba desde la salida analógica del módulo de **EM235**.

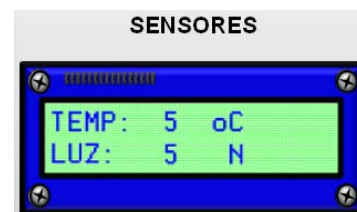


Funcionamiento del Hardware

13. Accione el interruptor, para que se ponga en funcionamiento; el PLC y el modulo de extensión EM235



14. Verifique que el PLC se encienda mediante el **Led de estado del PLC (STOP)**, de color anaranjado.
15. Cuando se encuentra corriendo algún programa se ilumina el led **RUN** de color verde.
16. Accione el pulsador, del visualizador de datos, para los **SENSORES**.
 - a) Verifique que se encienda el led rojo, indica que existe 5 voltios.
 - b) Verifique que se encienda el led verde, indica que está en funcionamiento el circuito electrónico.
17. Al accionar el pulsador, también se enciende el Lcd, para mostrar los datos de temperatura y luz, en el caso de los sensores.
18. Accione el pulsador, del visualizador de datos del **ACTUADOR**.
 - a) Verifique que se encienda el led rojo, indica que existe 5 voltios.
 - b) Verifique que se encienda el led verde, indica que está en funcionamiento el circuito electrónico.



19. Al accionar el pulsador también se encienden el Lcd, para mostrar el dato de la calefacción en el caso del actuador.



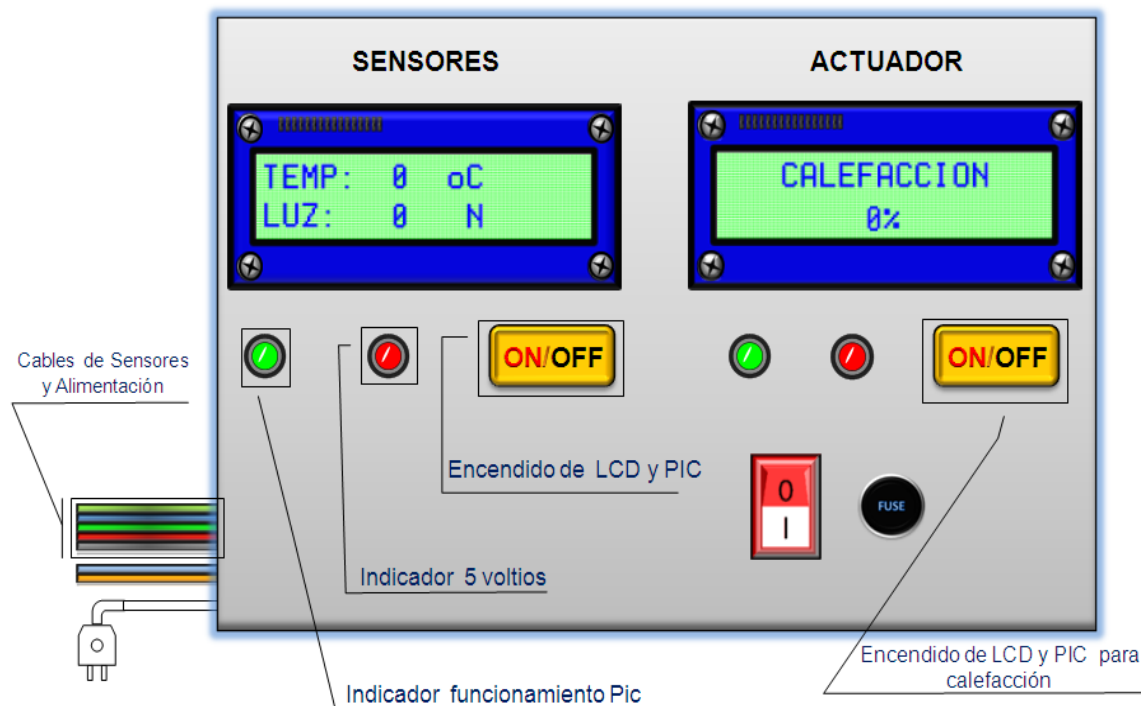
20. Accione el interruptor, para el funcionamiento de la niquelina.



21. El fusible ubicado junto al interruptor de la niquelina, sirve para proteger a la misma.



22. A continuación se muestra como quedaría en funcionamiento el visualizador de datos de los **SENSORES** y **ACTUADOR**.



23. Hasta el momento el hardware ya se encuentra en funcionamiento, en el caso de no ponerse en marcha algún elemento eléctrico, verifique las conexiones desde los pasos 1 al 22.

REQUISITOS PARA SOFTWARE:

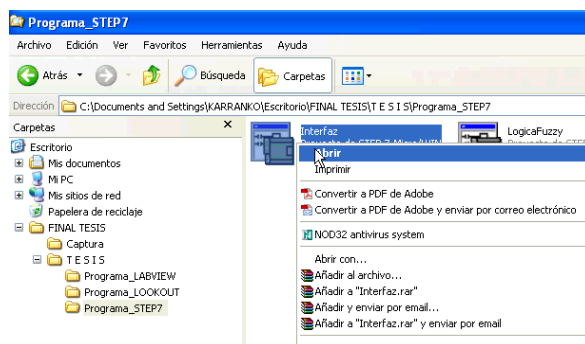
Para el funcionamiento del proyecto, debe tener instalado el siguiente software, con su respectiva licencia, para no tener ningún problema en la obtención o envío de datos de los sensores o actuador respectivamente.

- STEP 7-MicroWIN V4.0.1.10 o versión superior.
- National Instruments LabVIEW 7.1 o versión superior, con la librería Serial.
- National Instruments Lookout 5.0 o versión superior.
- Microsoft Office Excel 2003 o versión superior.

Labview realiza Lógica Difusa y Step 7 Transmite y Recibe Datos

Si se desea que Labview realice Lógica Difusa y el PLC sirva como interface, para transmitir y recibir datos, realice el siguiente procedimiento.

24. En la carpeta **Programa_STEP7**, seleccione el archivo **Interfaz.mwp** y clic en **Abrir**.

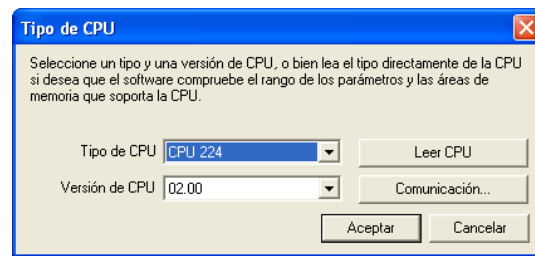


25. Antes de proceder, debe haber realizado los pasos del 13 al 23.

26. En la barra de Menú seleccione la opción **CPU** y **Tipo...**



27. Aparecerá una ventana **Tipo de CPU**, donde se debe seleccionar la opción del **Tipo de CPU** y **Versión de CPU**, este procedimiento es de forma manual.

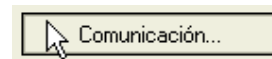


28. En caso de **no** conocer el **Tipo de CPU** y la **Versión de CPU**.

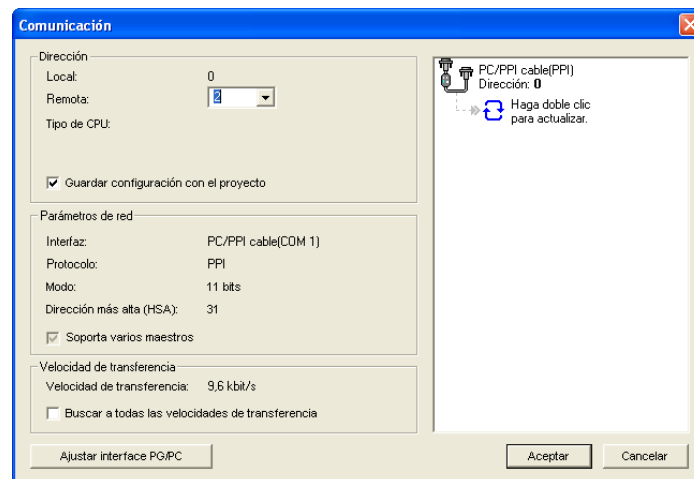
29. Hacer clic en **Leer CPU**, automáticamente se reconoce la información requerida.



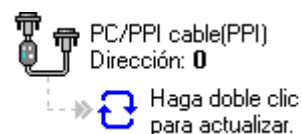
30. Haga clic en **Comunicación.**, para comunicarse con el PLC.



31. Aparece una ventana para actualizar la interfaz.



32. **Haga doble clic para actualizar**, sobre el texto que le pide realizar.



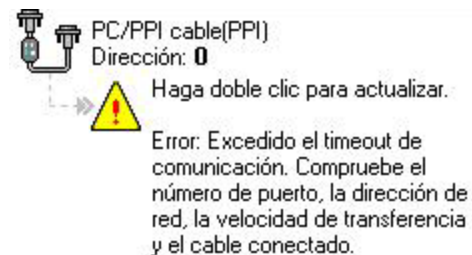
33. Comienza la búsqueda de las direcciones desde 0 a 126



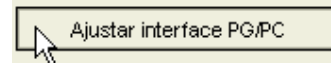
34. Si todo esta correcto, aparecerá el CPU con la respectiva dirección.



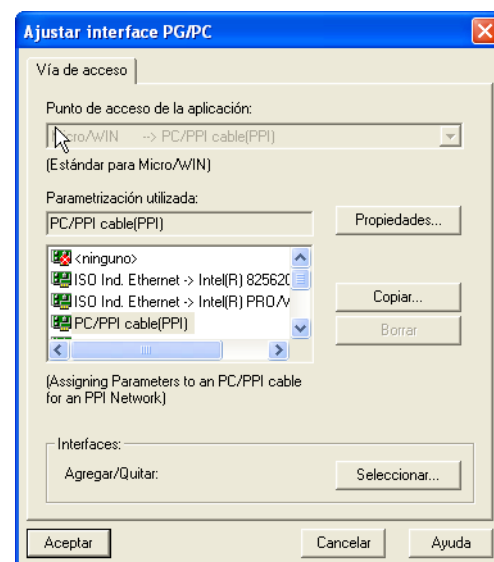
35. En caso de no estar correcto, aparecerá un error.



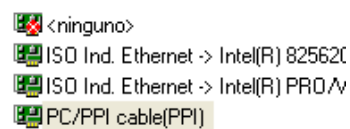
36. Pasos para identificar el error ocurrido.
- Verificar que el PLC este encendido. (Pasos desde el 13 al 23)
 - Comprobar el cable de comunicación esté conectado correctamente al PC. (Paso 2).
 - Comprobar el cable de comunicación esté conectado correctamente al PLC. (Paso 3)
 - Verificar el Tipo de CPU seleccionado. (Pasos desde del 26 al 34)
37. Haga clic en **Ajustar interface PG/PC**.



38. Aparece la ventana **Ajustar interface PG/PC**, para configurar el cable PC/PPI.



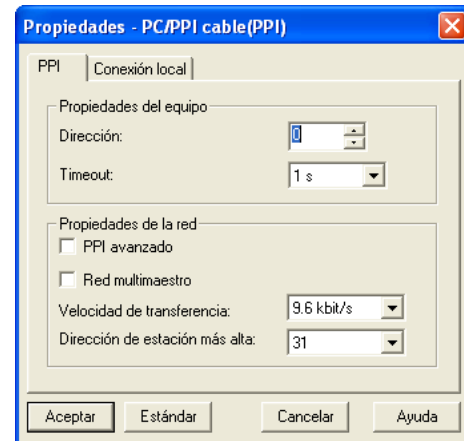
39. Clic en **PC/PPI cable (PPI)**.



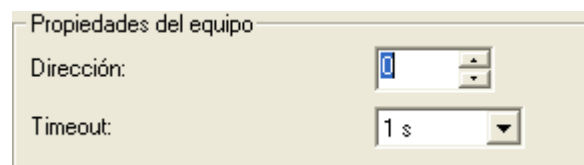
40. Haga clic en **Propiedades**



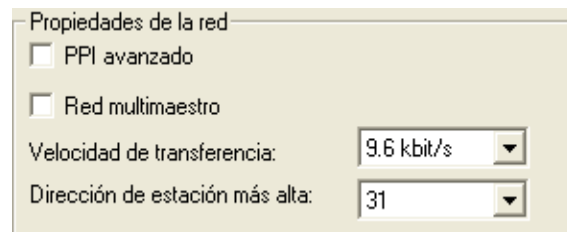
41. Aparece la ventana
Propiedades PC/PPI cable (PPI)



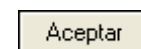
42. Verificar en *Propiedades del equipo*, la **Dirección** este en 0 y **Timeout** en 1s.



43. Verificar en *Propiedades de la red*, la **Velocidad de transferencia** en 9.6kbit/s y la **Dirección de estación más alta**.



44. Después de haber realizado correctamente la configuración, y no tener ningún problema, clic en **Aceptar** en cada ventana.

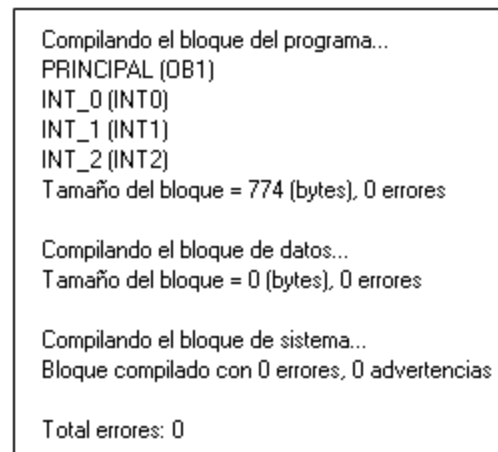


45. Luego de haber realizado todo el proceso anterior.

46. Buscamos en la Barra Estándar **Compilar** o **Compilar todo**, para verificar si esta correctamente el proyecto.



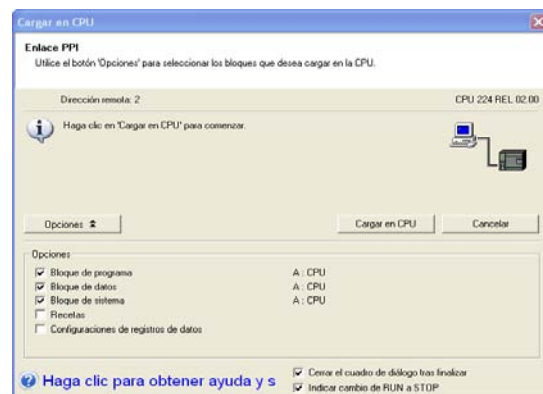
47. En la Ventana de resultados, aparece un mensaje si existe o no errores en el proyecto.



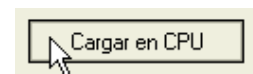
48. Haga clic en **Cargar CPU**, ubicando el botón en la barra Estándar



49. Si no existe error de comunicación aparece la siguiente ventana.

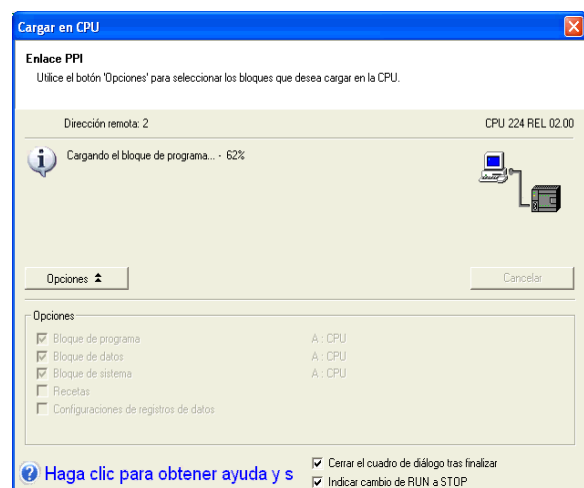


50. Haga Clic en el **Cargar en CPU**

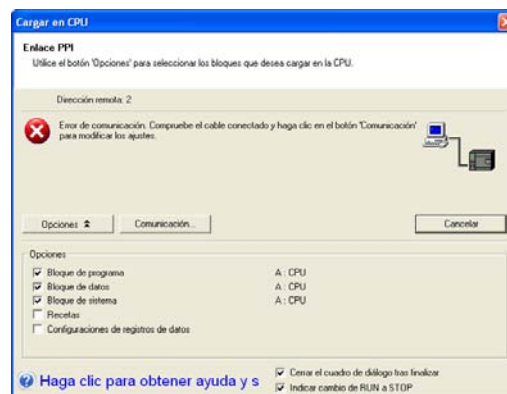


51. A continuación comienza a cargar:

- Bloque de programa.
- Bloque de Datos.
- Bloque de Sistema.



52. Si en el momento de hacer clic en Cargar en CPU, aparece una ventana indicando error de comunicación. Realice los pasos desde 36 al 44.



53. Si no tiene ningún problema al cargar el proyecto al CPU. Se cerrara automáticamente la ventana, quedando así cargado el proyecto en el CPU.
54. Ejecutar el proyecto.

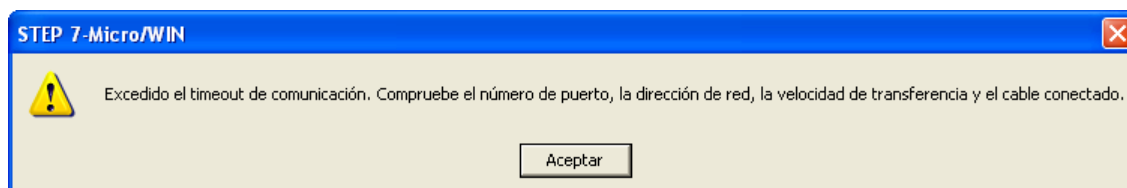
55. Haga clic en RUN



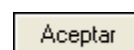
56. Después haga clic en **Estado del programa**, para comprobar el protocolo Freeport.



57. Aparece la siguiente ventana.



58. Haga clic en **Aceptar**.

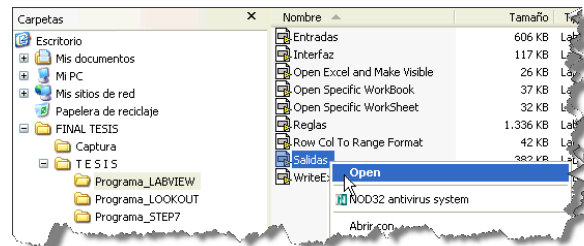


59. Verificar que el Led de RUN, del PLC se cambie a color verde



60. Desde este momento el protocolo Freeport se encuentra en funcionamiento.
61. Si desea puede cerrar Step 7.
62. Copiar el archivo **conexión.xls** en la raíz C:\

63. En la carpeta **Programa_LABVIEW**, seleccione el archivo **Salidas.vi**, haga clic derecho y haga clic en **Open**.



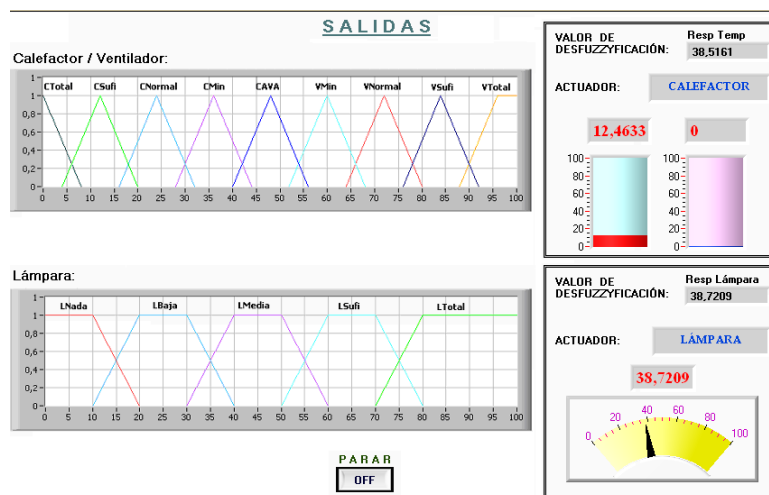
64. Haga clic en Run.



65. Cambia de forma el icono.



66. El sistema en este momento, ya esta adquiriendo los datos, de los sensores y enviar orden a los actuadores, a través del PLC.



UNIVERSIDAD POLITÉCNICA SALESIANA
FACULTAD DE INGENIERÍA EN SISTEMAS
TESIS DE GRADO

JOHANNA CELI SÁNCHEZ
CARLOS CARRANCO QUIÑONEZ

67. Simultáneamente se abrirá el archivo **conexión.xls**, mostrando las variables: Temperatura, calefacción, extractores, Luz.

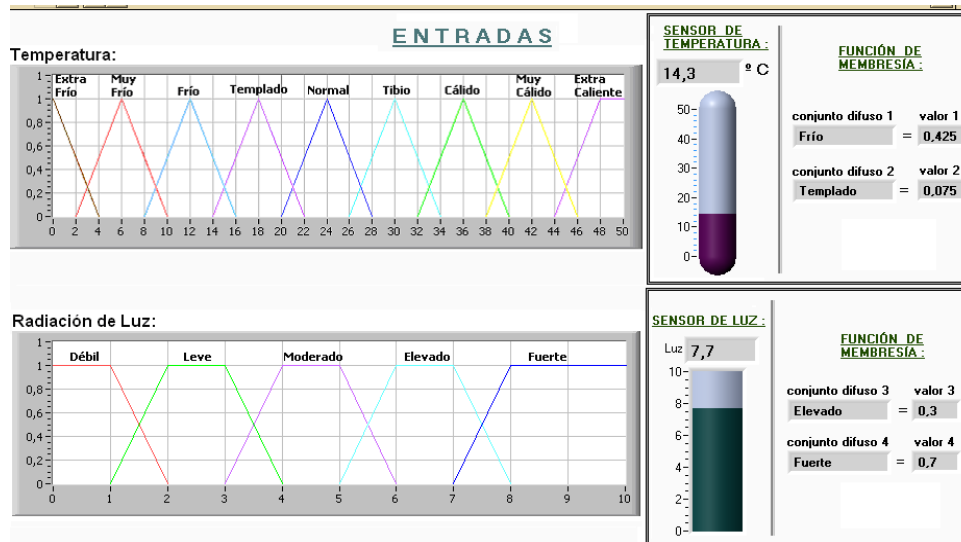
REGISTRO DE DATOS EN TIEMPO REAL:

SENSORES:	ACTUADORES:
Temperatura = 27,78	Ventilador = 13,7411
Luz = 5,43	Calefactor = 0
	Lámpara = 36,2641

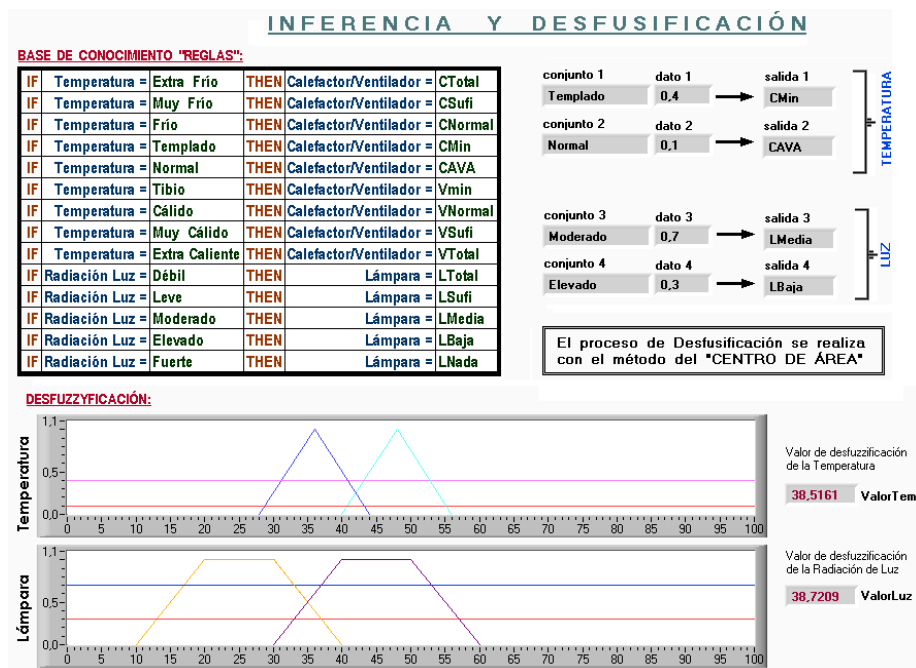
68. Para detener el proceso. Volver a la Labview en la interface **Salidas.vi** y haga clic en **PARAR**.

PARAR
OFF

69. Para supervisar las entradas.
- Abrir archivo **Entradas.vi**
 - Aparece ejecutándose archivo **Entradas.vi**.

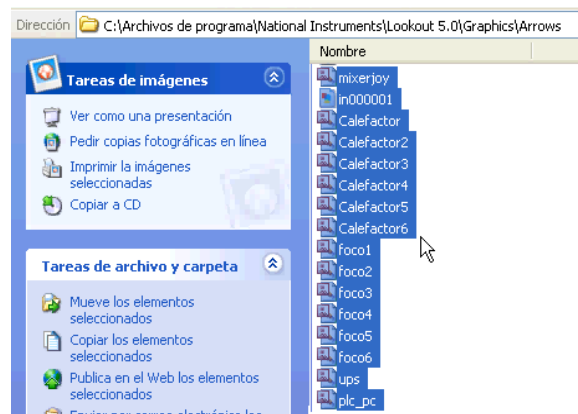


70. Para ver la base de conocimiento.
- Abrir archivo **Reglas.vi**
 - Aparece ejecutándose archivo **Reglas.vi**

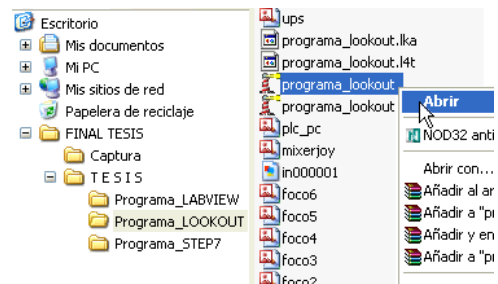


71. SUPERVISIÓN EN LOOKOUT.
72. Copiar todas las imágenes que están dentro de la carpeta **Programa_LOOKOUT**, en la siguiente dirección.

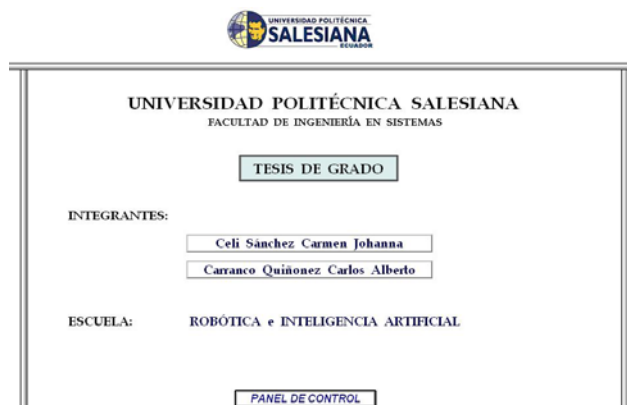
C:\Archivos de programa\National Instruments\Lookout
5.0\Graphics\Arrows



73. En la carpeta **Programa_LOOKOUT**, seleccione el archivo **programa.lks**, haga clic derecho y haga clic en **Abrir**.



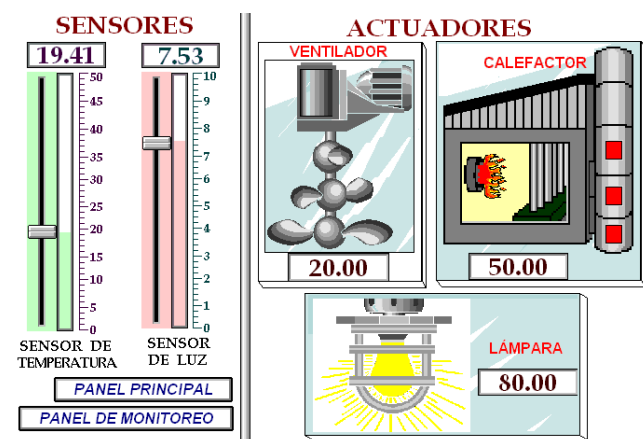
74. Aparece la siguiente interface



75. Haga clic en **PANEL DE CONTROL**.



76. Aparece una ventana, indicando las variables de sensores y el estado de los actuadores.

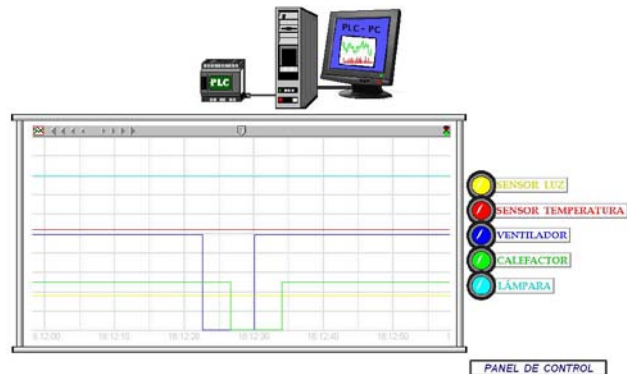


MONITOREO EN LOOKOUT

77. Haga clic en **PANEL DE MONITOREO**.

PANEL DE MONITOREO

78. Aparece una ventana, mostrando en un HyperTrend datos históricos de sensores y actuadores.



79. Con los botones
- **PANEL DE CONTROL.**
 - **PANEL DE MONITOREO.**
 - **PANEL PRINCIPAL.**

PANEL DE CONTROL

PANEL DE MONITOREO

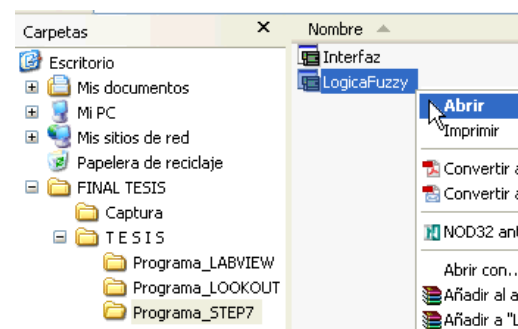
PANEL PRINCIPAL

Se puede navegar en los tres paneles, dependiendo el que se desee visualizar.

Step 7 realiza Lógica Difusa y Labview recibe datos.

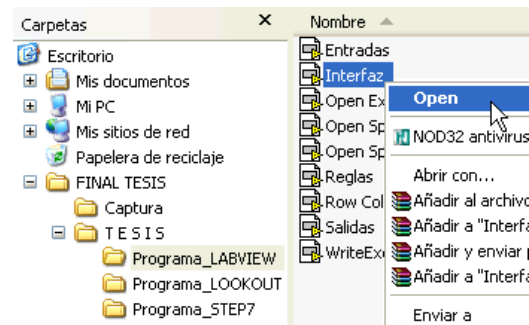
Si se desea que el PLC realice Lógica Difusa y Labview sirva como interface, realice el siguiente procedimiento.

80. En la carpeta **Programa_STEP7**, seleccione el archivo **LogicaFuzzy.mwp** y haga clic en **Abrir**.

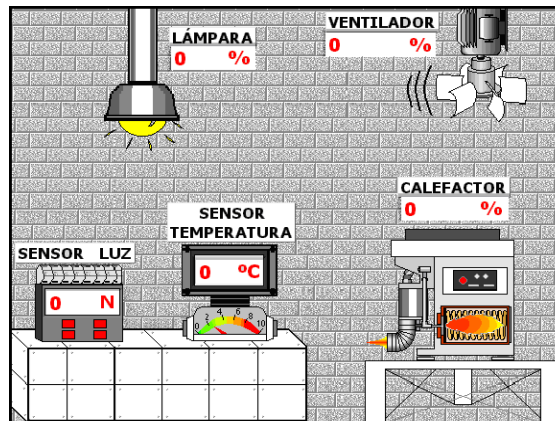


81. Luego siga los pasos del 25 hasta el 61.

82. En la carpeta **Programa_LABVIEW**, seleccione el archivo **Interfaz.vi** haga clic derecho y haga clic en **Open**.



83. Aparece la siguiente ventana, visualizándose: sensores, actuadores, con las respectivas variables.



84. Haga clic en Run.



85. Cambia de forma el icono.



86. El sistema ya está adquiriendo los datos de los sensores y enviando datos al PLC y este acciona los actuadores.

87. Simultáneamente se abrirá el archivo **conexión.xls**, mostrando las variables: Temperatura, calefacción, extractores, Luz.

UNIVERSIDAD POLITÉCNICA SALESIANA
FACULTAD DE INGENIERÍA EN SISTEMAS
TESIS DE GRADO

JOHANNA CELI SÁNCHEZ
CARLOS CARRANCO QUIÑONEZ

REGISTRO DE DATOS EN TIEMPO REAL:

SENSORES:		ACTUADORES:	
Temperatura =	27,78	Ventilador =	13,7411
Luz =	5,43	Calefactor =	0
		Lámpara =	36,2641

88. Para detener el proceso. Volver a la Labview en la interfaz y haga clic en **PARAR**.



89. Si desea realizar la supervisión en Lookout seguir los pasos 71 hasta 80.

5.3. PRUEBAS.

Primero se realiza las pruebas cuando el proceso de lógica difusa se efectúa en la computadora mediante LabView:

En las siguientes fotografías podemos observar que cuando la perturbación de la luz exterior es total, es decir una simulación de la luz del día a pleno sol, el sensor capta esta señal y por consiguiente el sistema responde apagando la lámpara dicróica, además cuando la temperatura es ambiente en 24°C, el termistor envía una señal y el sistema indica que no deben actuar ni la calefacción ni la ventilación, porque es el Set Point deseado:



Figura 5.2 Fotografía perturbación de luz al máximo.



Figura 5.3 Datos de sensores luz máxima y temperatura ambiente.

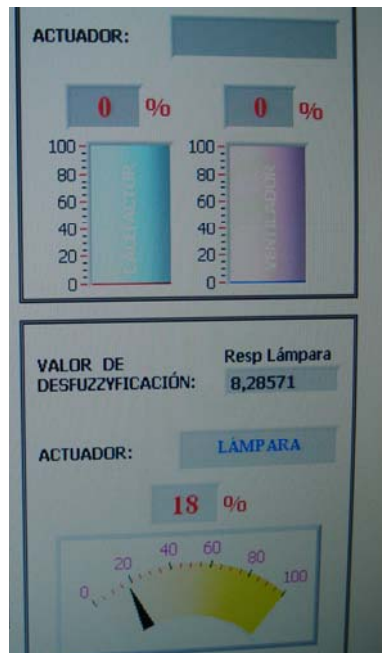


Figura 5.4 Luz al máximo y actuadores apagados (LabView).

La perturbación de luz se manipula a mayor grado, dando como respuesta el diodo encendido a 5V, paralelamente se distorsiona la temperatura con la ayuda de hielo para probar el sistema a 8°C, dando como resultado el calefactor accionado a 79%, ventilación al 0% y lámpara al 35%:



Figura 5.5 Fotografía de nivel 5V luz y 79% calefacción.



Figura 5.6 Lámpara a 5V.



Figura 5.7 Calefacción al 80% y lámpara al 35% (LabView).



Figura 5.8 Perturbación de temperatura con hielo.



Figura 5.9 Calefactor por convección funcionando al 80%.

La perturbación de luz se maneja a menor grado, dando como respuesta el diodo encendido a 8V, además se manipula la temperatura con la ayuda de agua caliente para probar el sistema a 29°C, dando como resultado el calefactor accionado a 0%, ventilación al 24%, y lámpara al 75%:



Figura 5.10 Temperatura a 29°C y luz a nivel 2.



Figura 5.11 Perturbación temperatura con agua caliente.



Figura 5.12 Ventilación al 24% y lámpara a 8v, 75% (LabView).



Figura 5.13 Dicroico funcionando a 8V.

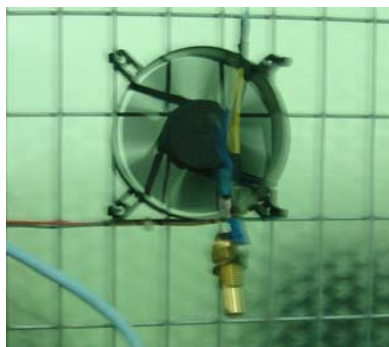


Figura 5.14 Ventilador girando a 5V.

La perturbación de luz se maneja a oscuridad, dando como respuesta el dicroico encendido a 12V, con un nivel de luz de 1:

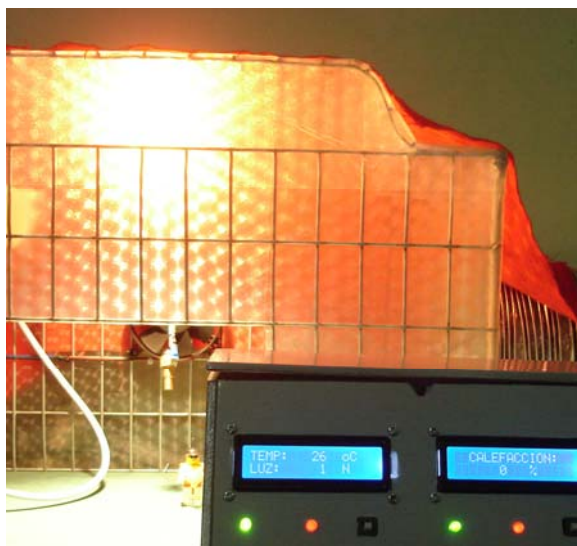


Figura 5.15 Lámpara funcionando a 12V.

En el segundo caso, cuando el proceso difuso se realiza en el PLC, las respuestas son muy similares a las explicadas anteriormente, la pequeña diferencia en los datos es de una cantidad, ya que en el PLC podemos operar con

números reales y escribirlos directamente, caso que no ocurre con LabView, ya que se debe transformar a bytes antes de la transmisión por lo que se genera pérdidas de información.

Las imágenes que varían son las interfaces en LabView, en el caso del hardware y perturbaciones, estas imágenes son las mismas:

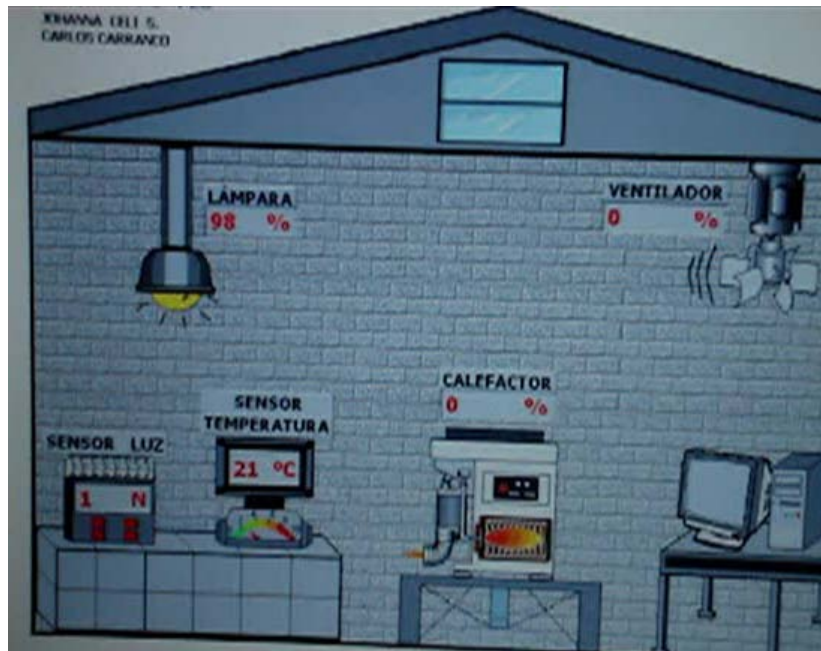


Figura 5.16 Fotografía del software de la interfaz en LabView cuando el PLC realiza Lógica Difusa, nivel de luz 1, lámpara al 98%, temperatura 21°C por lo que debe ser calefacción 0% y ventilación 0% (temperatura ideal).

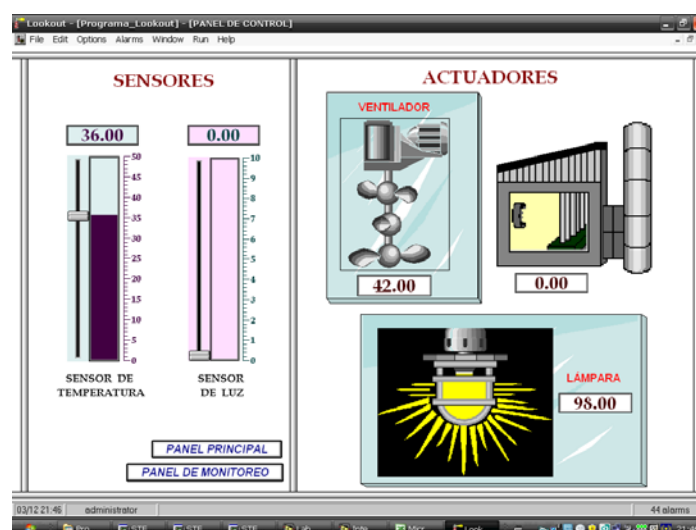


Figura 5.17 Software Lookout ejecutándose a 36°C, nivel de luz 0, ventilación al 42% y lámpara al 98%, calefacción 0%.

Para dar constancia que el sistema funciona correctamente, se disponen de un video grabado cuando el sistema se encuentra en ejecución, en ambos casos.

5.4. CONCLUSIONES Y RECOMENDACIONES.

5.4.1. CONCLUSIONES.

- Como conclusión principal, se confirma que mediante el sistema implementado, el proceso de lógica difusa es aplicable para los sistemas HMI / Scada.
- Los resultados obtenidos luego de realizar las respectivas pruebas con el sistema, fueron positivas. El hardware y el software implementados trabajan de la forma deseada.
- El proyecto de investigación puede ser implementado en varias áreas como: en la industria automotriz, en la industria aeroespacial, en la ciencia médica, en la industria de manufactura, en robótica, en el campo de seguridad, en domótica, en la automatización de industrias, en automatización de procesos en general, en sistemas HMI/SCADA, en la ganadería, en agronomía, invernaderos, en el control de temperatura para sistemas de calefacción y sistemas de enfriamiento o aire acondicionado, hornos, refrigeración, control de humedad, presión, control de luminarias, sistemas de aceleración, control de nivel, control de caudal, etc.
- Para obtener datos reales en forma analógica y poder procesar las variables con lógica difusa, se incluyó en el prototipo un módulo de extensión de entradas y salida analógicas EM235, debido a que el PLC Siemens Simatic S7-200 de CPU 224, trabaja de forma digital.
- El módulo EM235 posee 4 entradas analógicas y 1 salida analógica, debido a esto, fue necesario activar los actuadores restantes (ventiladores y

lámpara) de forma digital con 4 estados lógicos a 2 bits (0v, 5v, 8v, y 12v), para mantener el concepto de lógica difusa.

- Los sistemas Scada trabajan generalmente con lógica clásica, la cual no brinda resultados completos, debido a que trabaja sólo con dos estados por ejemplo 0 y 1, encendido y apagado, 0 voltios y 5 voltios, 0 voltios y 12 voltios, etc. Con lógica difusa podemos cambiar estos 2 estados, por un rango más amplio, que se ajusta más a la realidad y a las necesidades del ser humano y de los sistemas de control.
- Por la facilidad de comunicación con el autómatas, y por una programación gráfica y orientada a la adquisición de datos, se hizo uso de LabView para el control, además se creó una interfaz de usuario en Lookout para la supervisión y monitoreo.
- Con los sistemas basados en lógica difusa se pueden valorar mayor cantidad de variables, como variables lingüísticas, no numéricas, simulando el conocimiento humano.
- Se relaciona entradas y salidas, sin tener que entender todas las variables, permitiendo que el sistema pueda ser más confiable y estable que uno con un sistema de control convencional.
- La utilización de lógica difusa es aconsejable para procesos muy complejos, cuando se carece de un modelo matemático simple o para procesos no lineales.
- Evitar el uso de lógica difusa, si el control convencional rinde un resultado favorable, o cuando existe un modelo matemático de fácil solución.

5.4.2. RECOMENDACIONES.

- Seleccionar una fuente eléctrica adecuada a la potencia y corriente que demandan de los actuadores y los circuitos en general.
- Antes de implementar un sistema, se debe estudiar el número y tipo de entradas y salidas tanto digitales como analógicas; con el fin de adquirir el autómata y módulos más adecuados para el manejo de las señales.
- Realizar una investigación previa, con respecto a los sensores que se han de implementar de acuerdo a la variable controlada del proceso.
- Seleccionar los actuadores adecuados con su respectiva alimentación eléctrica y potencia.
- Revisar el manual y data sheets de todos los elemento electrónicos, antes de su utilización, poniendo interés en la marca, modelo, serie y versión.
- Establecer la ubicación y la distancia de los sensores y actuadores con respecto al tablero de control, para definir si se trabaja con corriente o voltaje, ya que cuando se transmite por corriente a largas distancias la señal no se degrada, es inmune a la captación del ruido, por lo que brinda mayor confiabilidad.
- Se recomienda adquirir las licencias del software: LabView, Lookout y Step7, para implementar el sistema a nivel industrial o empresarial.
- Revisar el manual de usuario detallado en éste capítulo, antes de poner en funcionamiento el sistema.
- En el momento de reemplazar un elemento electrónico, realizar el cambio teniendo en cuenta las respectivas características, para evitar datos erróneos o un corto circuito.

REFERENCIAS BIBLIOGRÁFICAS

ARMESTO, ISAD_Tema7_2.pdf [en línea], Sensores y Actuadores Industriales, 17/09/2007, [Fecha de revisión 26 de Nov. del 2009] <http://tv.uvigo.es/uploads/material/Video/1709/ISAD_Tema7_2.pdf>

Penayo Marti Carlos María, Termoresist.pdf [en línea], SENSORES Y ACTUADORES, 18/04/2004, [Fecha de revisión 25 de Nov. del 2009], <<http://www.pol.una.py/archivos/IngEltrnik/Sensores/Termoresist.pdf>>

Dani meganeboy, Sensores en el automóvil - sensores de temperatura [en línea], Sensores en el automóvil, Actualizada: 15 Abril, 2008, [Fecha de revisión 25 de Nov. del 2009], <<http://www.mecanicavirtual.org/sensores7.htm>>

DETECCION Y CLASIFICACION AUTOMATICA DE PATRONES EN POLISOMNOGRAMAS DE LACTANTES Y NINOS APLICANDO LOGICA DIFUSA Y TECNICAS NEURO-DIFUSAS [en línea], DETECCION Y CLASIFICACION AUTOMATICA DE PATRONES EN POLISOMNOGRAMAS DE LACTANTES Y NINOS APLICANDO LOGICA DIFUSA Y TECNICAS NEURO-DIFUSAS, [Fecha de revisión 25 de Nov. del 2009] <<http://www.conicyt.cl/bases/fondecyt/proyectos/01/2003/1030545.html>>

DISEÑO DE UN TRANSMISOR DE TEMPERATURA CON Termistor [en línea], DISEÑO DE UN TRANSMISOR DE TEMPERATURA, 23 de Agosto de 2004, [Fecha de revisión 26 de Nov. del 2009], <<http://proton.ucting.udg.mx/dpto/maestros/mateos/clase/teoria/temperatura/thr/>>

Montoya Jaime, Ejemplos de diagramas UML, interfaces gráficas de usuario, y usos del UML en la ingeniería inversa, [en línea], 3 de Octubre de 2008, <<http://www.docstoc.com/docs/3636424/Ejemplos-de-diagramas-UML-interfaces-gr%C3%A1ficas-de-usuario-y-usos-del-UML-en-la-ingenier%C3%ADa-inversa>>

BALCELLS JOSEP, ROMERAL JOSÉ LUIS, Autómatas Programables, Autómatas Programables – Google Libros, [en línea], [Fecha de revisión 18 de Diciembre del 2009],

<<http://books.google.com.ec/book?id=xfSJJADgd70C&pg=PT307&dq=estandar+422&cd=2#v=onepage&q=bucle%20de%20corriente&f=false>>

Escola Universitària d'Informàtica Sabadell, Universitat Autònoma de Barcelona (U.A.B) [en línea], Diseño de aplicaciones SCADA con LabView Memoria del proyecto de Ingeniería Técnica en Informática de Sistemas 31 de Octubre de 2003, [Fecha de revisión 26 de Nov. del 2009]

<<http://personal.redestb.es/efigueras/memoria.htm>>

Estructuras [en línea], Estructuras, 09 de Noviembre de 2004, [Fecha de revisión 26 de Nov. del 2009],

<<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/estructuras.htm>>

MENÉNDEZ Díez FAUSTINO. Higiene Industrial: Manual para la formación del especialista, Higiene Industrial: Manual para la formación del especialista - Google Libros – [Fecha de revisión 18 de Diciembre del 2009], [en línea],

<http://books.google.com.ec/books?id=W8EnsQ2_q0cC&pg=PT423&dq=luminarias&cd=1#v=onepage&q=luminarias&f=false>

Funcionamiento [en línea], Funcionamiento, 09 de Noviembre de 2004, [Fecha de revisión 26 de Nov. del 2009],

<<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/FUNCIONAMIENTO/funcionamiento.htm>>

Fundamentos del control automático industrial [en línea], FUNDAMENTOS DEL CONTROL AUTOMÁTICO INDUSTRIAL, 08 de Julio de 2009, [Fecha de revisión 26 de Nov. del 2009] <http://www.sapiensman.com/control_automatico/index.htm>

Fuzzy Set., Subconjuntos Borrosos [en línea]. 10 de abril de 2007 2002 [Fecha de revisión 25 de Nov. del 2009]. Disponible en: <http://www.dei.uc.edu.py/tai2000/logica/logica.htm> >

Fvelasco, [conecte_ni_labview_a_cualquier_red_industrial_y_plc.pdf](#) [en línea], Conecte LabVIEW a Cualquier Red Industrial y PLCs, 114/11/07, [Fecha de revisión 26 de Nov. del 2009], ftp://ftp.ni.com/pub/branches/latam/nidays_2007/conecte_ni_labview_a_cualquier_red_industrial_y_plc.pdf>

Haciendo sensores, Cómo funcionan los sensores en la tarjeta GoGo [en línea], 08/24/2007, [Fecha de revisión 25 de Nov. del 2009], <http://www.gogoboard.org/cocoon/gogosite/documentation/makingSensors.xsp?!ang=es>>

Instrumentacion industrial – Simbología [en línea], [Instrumentación industrial – Normas de representación de los instrumentos – Simbología](#), 07 de Julio de 2009, [Fecha de revisión 25 de Nov. del 2009], http://www.sapiensman.com/control_automatico/control_automatico8.htm>

Jorge Falcó, [siesenso.pdf](#) [en línea], Sensores e Instrumentación Electrónica, 25/05/2000, [Fecha de revisión 26 de Nov. del 2009], http://www.cps.unizar.es/~te/Docencia_archivos/senins_archivos/apuntes/siesenso.pdf>

Joseph Vilalta Marzo, [TRAC_introUML.pdf](#) [en línea], Introducción a UML, 16/01/2008, [Fecha de revisión 26 de Nov. del 2009], http://www.vico.org/aRecursosPrivats/TRAD_introUML.pdf>

La Nueva España, La Nueva España - Diario Independiente de Asturias - Cuencas - La lógica difusa ayudará a predecir enfermedades mediante el estudio

de los genes, La lógica difusa ayudará a predecir enfermedades mediante el estudio de los genes [en línea], 13 de julio de 2007, [Fecha de revisión 26 de Nov. del 2009]

<http://www.lne.es/secciones/noticia.jsp?pRef=1697_38_539227_Cuencas-logica-difusa-ayudara-predecir-enfermedades-mediante-estudio-genes>

Pillajo Angos Carlos, CtrlProcesos.pdf [disco compacto], Control de Procesos, 04/Enero/2005. (CD facilitado por el Ing. Carlos Pillajo, profesor de la Asignatura: Sistemas Scada de 8^{vo} Robótica e Inteligencia Artificial, Facultad de Ingeniería en Sistemas, Universidad Politécnica Salesiana).

PLC [en línea], El PLC, 09 de Noviembre de 2004, [Fecha de revisión 26 de Nov. del 2009],

<<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm>>

Practica1_siemens.pdf [en línea], SIMATIC S7-200, Modificado 19/10/2007, [Fecha de revisión 26 de Nov. del 2009],
<http://isa.umh.es/asignaturas/asct/practicas_automatas/practica1_siemens.pdf>

p10.pdf [en línea], PRÁCTICA 10, CONTROL DE UN MOTOR DC MEDIANTE UN AUTÓMATA PROGRAMABLE (PLC), 19/05/2009, [Fecha de revisión 26 de Nov. del 2009], <<http://isa.umh.es/asignaturas/ai/practicas/p10.pdf>>

Robert J. Stephenson et. al, temperatura.pdf [en línea], Temperature Measurement, 11/09/03, [Fecha de revisión 26 de Nov. del 2009]

<<http://proton.ucting.udg.mx/dpto/maestros/mateos/clase/teoria/temperatura/pdf/temperatura.pdf>>

Santiago Orellana, InstCont04.pdf [en línea], INSTRUMENTACION Y CONTROL, 07/06/2004, [Fecha de revisión 26 de Nov. del 2009],

<<http://www.highlights.com.ec/docs/InstCont04.pdf>>

Santiago Orellana, IA_SACADA_2008 [en línea], Automatización Industrial y Sistemas HMI – SCADA, 05/03/2008, [Fecha de revisión 26 de Nov. del 2009], <http://www.highlights.com.ec/docs/IA_SCADA_2008.pdf>

Sergio Llenares, automatas.pdf [en línea], Manual usuario Simatic S7-200 CPU 224, 06/04/2006, [Fecha de revisión 26 de Nov. del 2009] <<http://www.salesianos.edu/alcoy.juanxxiii/dpts/docs/automatas.pdf>>

Sistemas Inteligentes [en línea], Unidad Temática 3: PRINCIPIOS DE CONTROL Y AUTOMACION, 10 de Diciembre de 2007, [Fecha de revisión 26 de Nov. del 2009], <<http://www.profesores.frc.utn.edu.ar/industrial/sistemasinteligentes/>>

Technical Documents - Documentos Técnicos [en línea], Instrumentación Industrial, Technical English – Spanish Vocabulary, 20 de Julio de 2009, [Fecha de revisión 26 de Nov. del 2009] <<http://www.sapiensman.com/ESDictionary/docs/d17.htm>>

TEO, practica1_s7200.pdf [en línea], Practica 1 INtroducción al autómeta Siemens s7-200, 16/10/2006, [Fecha de revisión 26 de Nov. del 2009], <http://isa.umh.es/asignaturas/asct/practicas_automatas/practica1_s7200.pdf>

Transductores y Sensores en la Automatización Industrial - Apuntes de Ingeniería Industrial [en línea], Transductores y Sensores en la Automatización Industrial, 26 de Noviembre de 2009, [Fecha de revisión 25 de Nov. del 2009], <http://www.elprisma.com/apuntes/ingenieria_industrial/transductoressensores/default.asp>

valfaro, Introducción a los sistemas de control supervisor y de adquisición de datos (SCADA) [en línea], infpPLC_net_Introduccion_Sistemas_SCADA.pdf , 16/02/2004, [Fecha de revisión 26 de Nov. del 2009], <http://www.infoplc.net/Documentacion/Docu_SCADA/infpPLC_net_Introduccion_Sistemas_SCADA.pdf>

Webmaster, Scadas, SISTEMAS SCADA [en línea], Última modificación: 02/03/2006, [Fecha de revisión 26 de Nov. del 2009], <<http://www.automatas.org/redes/scadas.htm>>

Zulma Janet Hernandez Paxtian, TUTORIAL DE LOGICA DIFUSA Y SUS APLICACIONES EN CONTROL [en línea]., 03 de junio de 2002 [Fecha de revisión 25 de Nov. del 2009]. <<http://kim.ece.buap.mx/posgrado/Fuzzytutorial/>>

10 Razones para Agregar LabVIEW a su PLC - Developer Zone - National Instruments [en línea], 10 Razones para Agregar LabVIEW a su PLC, 26/11/2006, [Fecha de revisión 26 de Nov. del 2009], <<http://zone.ni.com/devzone/cda/tut/p/id/6329>>

¿QUÉ ES EL SISTEMA SCADA? [en línea], - Véalo en QuimiNet.com: 15/10/2006, [Fecha de revisión 25 de Nov. del 2009]. Disponible en: <http://www.quiminet.com/ar3/ar_vcdadvcaasdRsDF-que-es-el-sistema-scada.htm>

Zepeda Peña (Revisado por Ing. Carlos Pillajo A.), Simbolos de Instrumentación.pdf [cd room], SIMBOLOS DE INSTRUMENTACION, 04/Enero/2005. (CD facilitado por el Ing. Carlos Pillajo, profesor de la Asignatura: Sistemas Scada de 8^{vo} Robótica e Inteligencia Artificial, Facultad de Ingeniería en Sistemas, Universidad Politécnica Salesiana).

ANEXOS

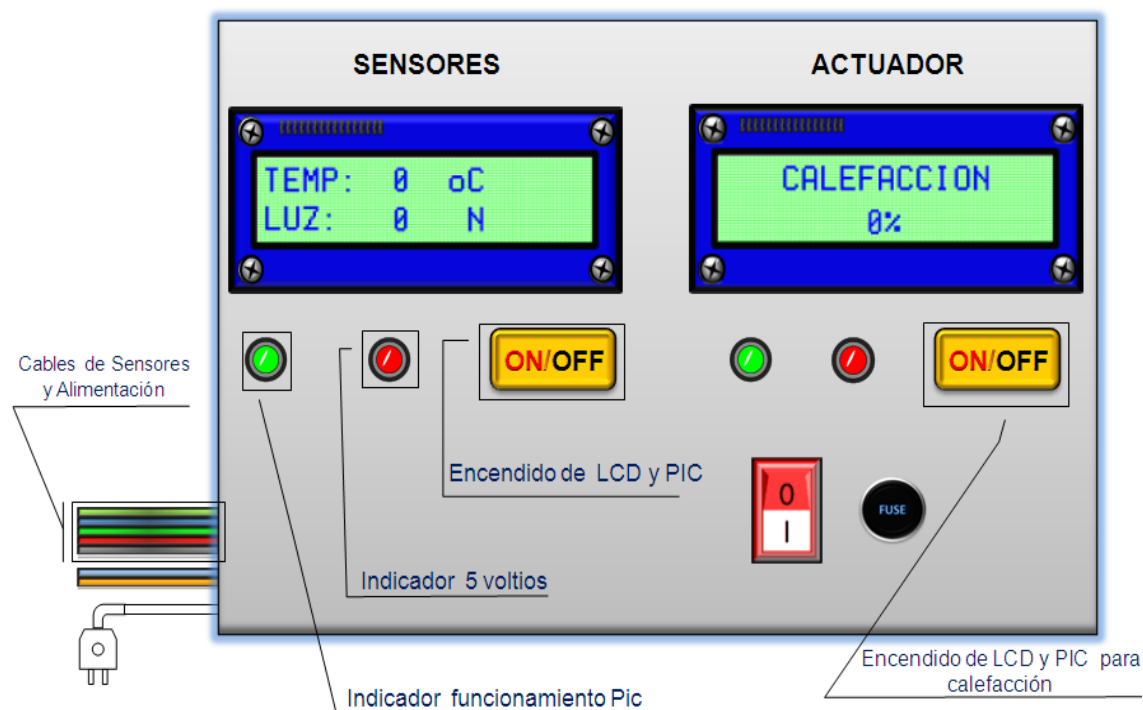
ANEXO 1. VISUALIZADOR DE DATOS

En el anexo 1 se presenta el visualizador de datos, ya que es una parte opcional del proyecto de tesis, se lo puede o no adjuntar sin que influya en el propósito general.

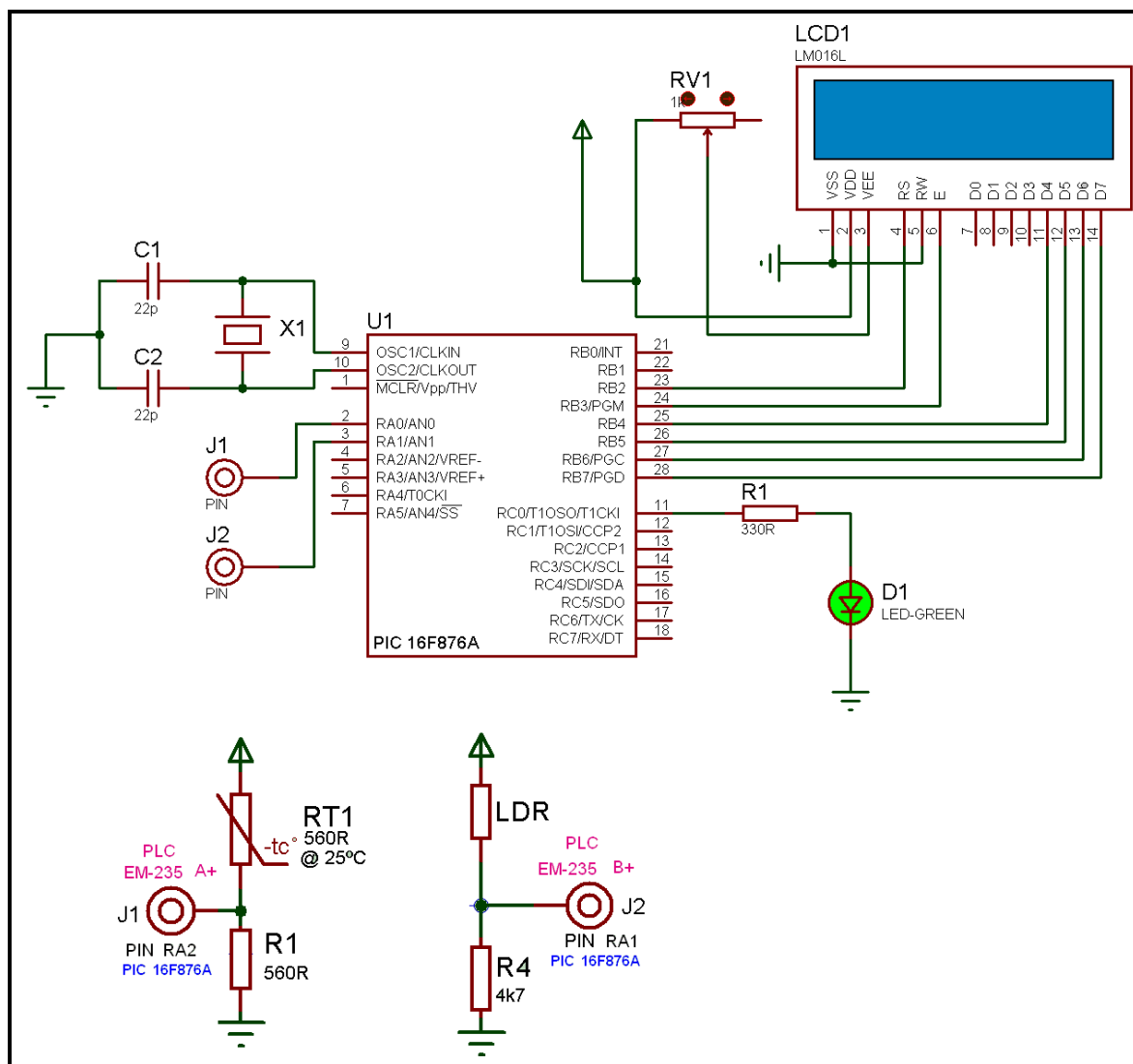
Este visualizador nos muestra, en el primer LCD, los valores de los 2 sensores: temperatura y luz que van a conectarse con el Módulo EM235 como entradas analógicas; en el segundo LCD nos muestra el valor de la salida analógica de la Calefacción que nos entrega el módulo.

También consta de Leds indicadores: Verdes para confirmar que los microcontroladores (PIC's) funcionan correctamente, Rojos para indicar la existencia de fuente eléctrica (+5Vcc). Posee pulsadores para el encendido y apagado de los microcontroladores.

Además está constituido de un fusible y un switch para seguridad de la conexión a potencia del calefactor (110 Vac).



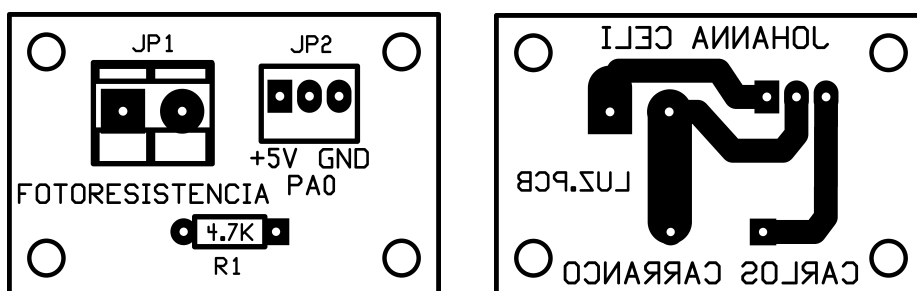
Primer diagrama para LCD de sensores con PIC 16 F 876A:



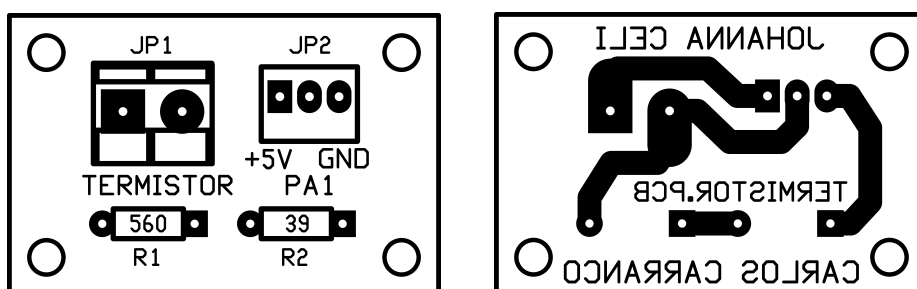
Lista de elementos de la placa de sensores:

Item	Cantidad	Referencia	Descripción
1	1	C1	0.1uF
2	1	C2	100uF
3	1	C3	10uF
4	2	C4,C5	22pF
5	1	D1	1N4007
6	2	D3,D2	LED
7	3	JP1,JP3,JP5	HEADER 2
8	2	JP4,JP2	HEADER 3
9	3	JP6,JP7,JP8	HEADER 4
10	1	P1	10K
11	1	R1	100K
12	2	R3,R2	470
13	1	R4	10
14	1	S1	SW SPDT
15	1	S2	SW PUSHBUTTON
16	1	U1	PIC16F876A
17	1	U2	LNT-211
18	1	XT1	4MHz

Placa por separado del sensor de luz:



Placa por separado del sensor de temperatura:



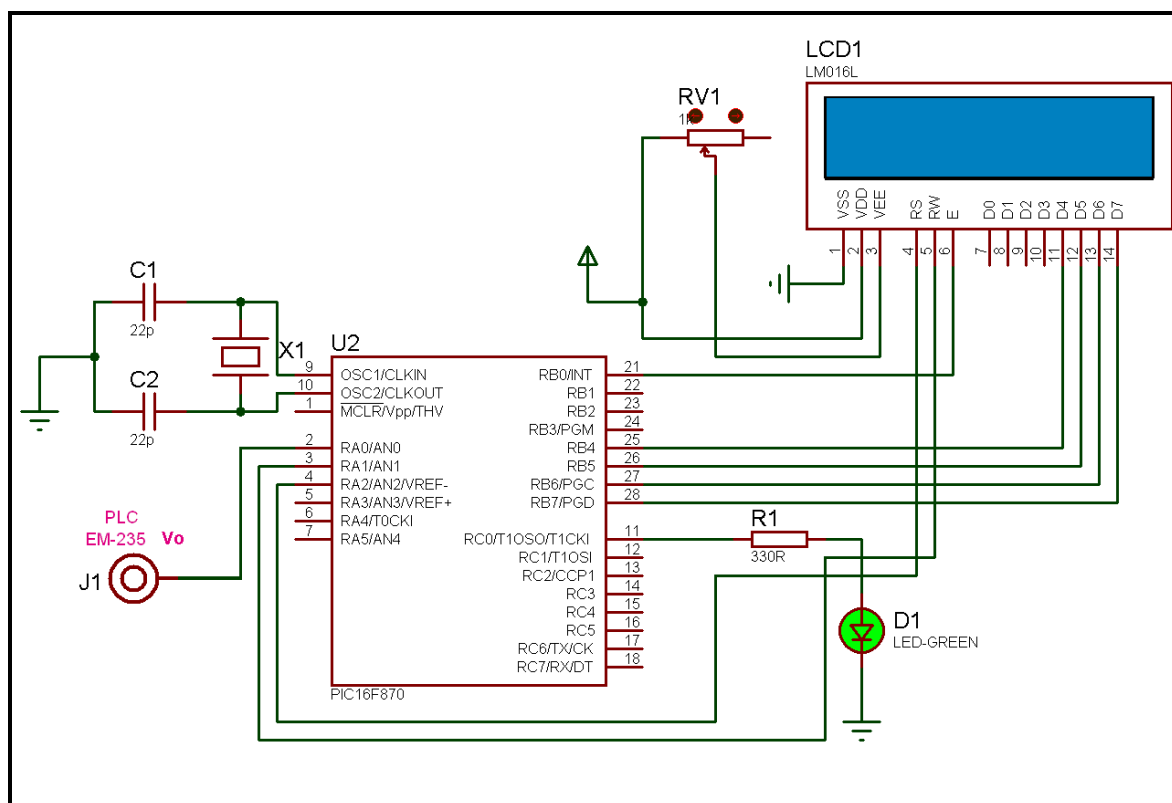
Lista de elementos del sensor fotorresistencia:

Item	Cantidad	Referencia	Descripción
1	1	JP1	HEADER 2
2	1	JP2	HEADER 3
3	1	R1	470

Lista de elementos del sensor termistor:

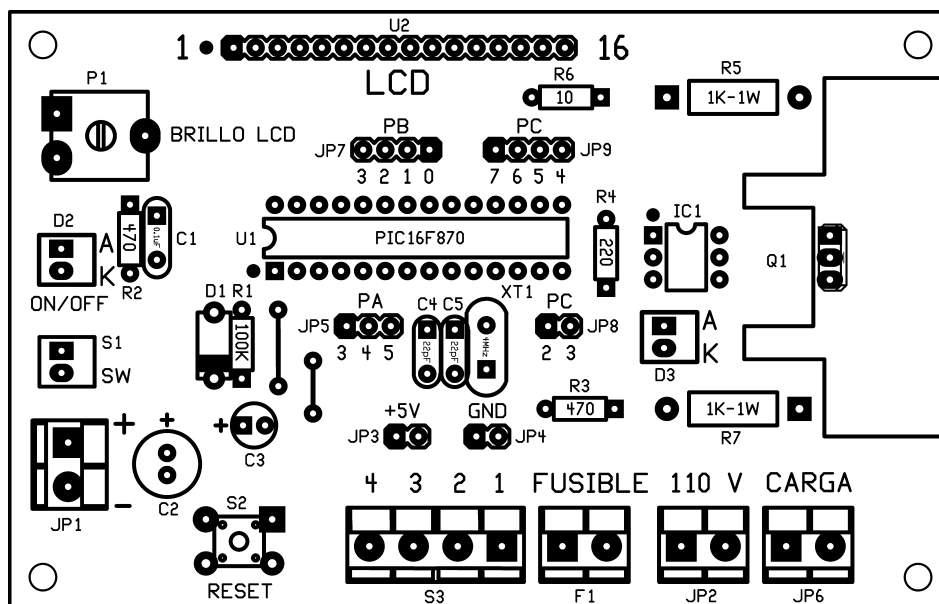
Item	Cantidad	Referencia	Descripción
1	1	JP1	HEADER 2
2	1	JP2	HEADER 3
3	1	R1	560
4	1	R2	39

Segundo diagrama para el control de la calefacción, con LCD y PIC 16F870:

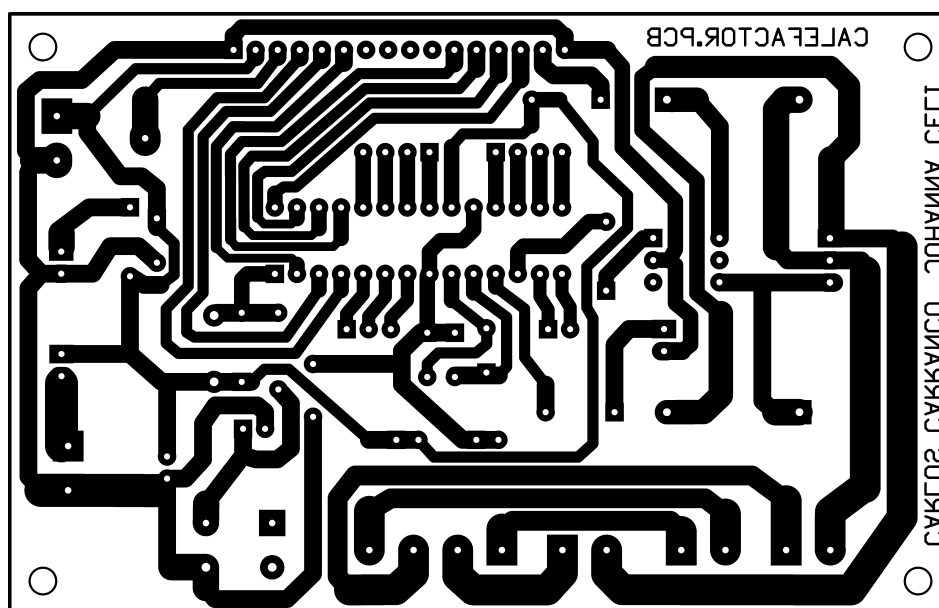


Placa PCB:

Vista frontal de placa para la calefacción, Lcd y Pic 16F870.



Vista posterior de placa para la calefacción, Lcd y Pic 16F870.



Lista de elementos para la placa de la calefacción:

Item	Cantidad	Referencia	Descripción
1	1	C1	0.1uF
2	1	C2	100uF
3	1	C3	10uF
4	2	C5,C4	22pF
5	1	D1	1N4007
6	2	D2,D3	LED
7	1	F1	FUSE
8	1	IC1	MOC3010
9	5	JP2,JP3,JP4,JP8,JP11	HEADER 2
10	1	JP5	HEADER 3
11	1	JP6	CARGA(R)
12	2	JP9,JP7	HEADER 4
13	1	P1	10K
14	1	Q1	TRIAC TIC226
15	1	R1	100K
16	2	R2,R3	470
17	1	R4	220
18	2	R5,R7	1K-1W
19	1	R6	10
20	1	S1	SW SPDT
21	1	S2	SW PUSHBUTTON
22	1	S3	SW DPST
23	1	U1	PIC16F870
24	1	U2	LNT-211
25	1	XT1	4MHz

Código fuente para el microcontrolador 16F876A para el sensor de temperatura y sensor de luz, realizado en MikroBasic.

!*****

!* Name : Sensores
 !* Author : Johanna Celi y Carlos Carranco
 !* Date : 30/07/2009
 !* Notes : PIC 16F876A

!*****

```

program termistor
dim temp as float
dim temperatura as word
dim temp3 as word
dim luz3 as word
dim luz as word
dim luz2 as float
dim tempout as string[2]
dim luzout as string[2]
```

MAIN:

```

TRISA=%11111111
TRISB=%00000000
TRISC=%00000000
Lcd_Config(PORTB,7,6,5,4,PORTC,6,5,4)
Lcd_Cmd(Lcd_CURSOR_OFF)
ADCON1=0
WHILE TRUE
temperatura = Adc_Read(2)
luz = Adc_Read(3)
temp = (((temperatura*50)/1024)-13.223)/0.4839
luz2 = ((luz+102.4)*10)/1024
temp3 = word(temp)
luz3 = word(luz2)
Lcd_Out (1,1,"TEMP:")
Lcd_Out (2,1,"LUZ:")
Lcd_Out (1,13,"oC")
Lcd_Out (2,13,"N")
WordToStr(temp3, tempout)
Lcd_Out (1,6,tempout)
WordToStr(luz3, luzout)
Lcd_Out (2,6,luzout)
PORTC=128
Delay_ms(1000)
wend
END.

```

Código fuente para el microcontrolador 16F870 para la calefacción, realizado en MicroCode Studio.

```

!*****
!* Author : Johanna Celi y Carlos Carranco
!* Notice : Calefacción
!* Date : 30/07/2009
!* Notes : PIC 16F870
!*****

TRISA=%00000001
TRISB=%00000000
TRISC=%00000000
ADCON1=%0001110
DEFINE OSC 4
DEFINE ADC_BITS 8
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPLEUS 50
DEFINE LCD_DREG PORTB
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTA
DEFINE LCD_RSBIT 2

```



```

DEFINE LCD_RWREG PORTA
DEFINE LCD_RWBIT 1
DEFINE LCD_EREG PORTB
DEFINE LCD_EBIT 0
DEFINE LCD_BITS 4
DEFINE LCD_LINES 2
CALF VAR BYTE
PAUSE 300
LAZO:
ADCIN 0, CALF
CALF = (CALF*100)/256
LCDOUT $FE,1
LCDOUT $FE,$80+2,"CALEFACCION:"
LCDOUT $FE,$C0+10,"% "
LCDOUT $FE,$C0+6, DEC CALF
IF CALF >= 10 THEN
    HIGH PORTC.1
ENDIF
IF CALF <=9 THEN
    LOW PORTC.1
ENDIF
HIGH PORTC.0
PAUSE 300
GOTO LAZO
END

```

ANEXO 2. FOTOGRAFÍAS DEL SISTEMA



